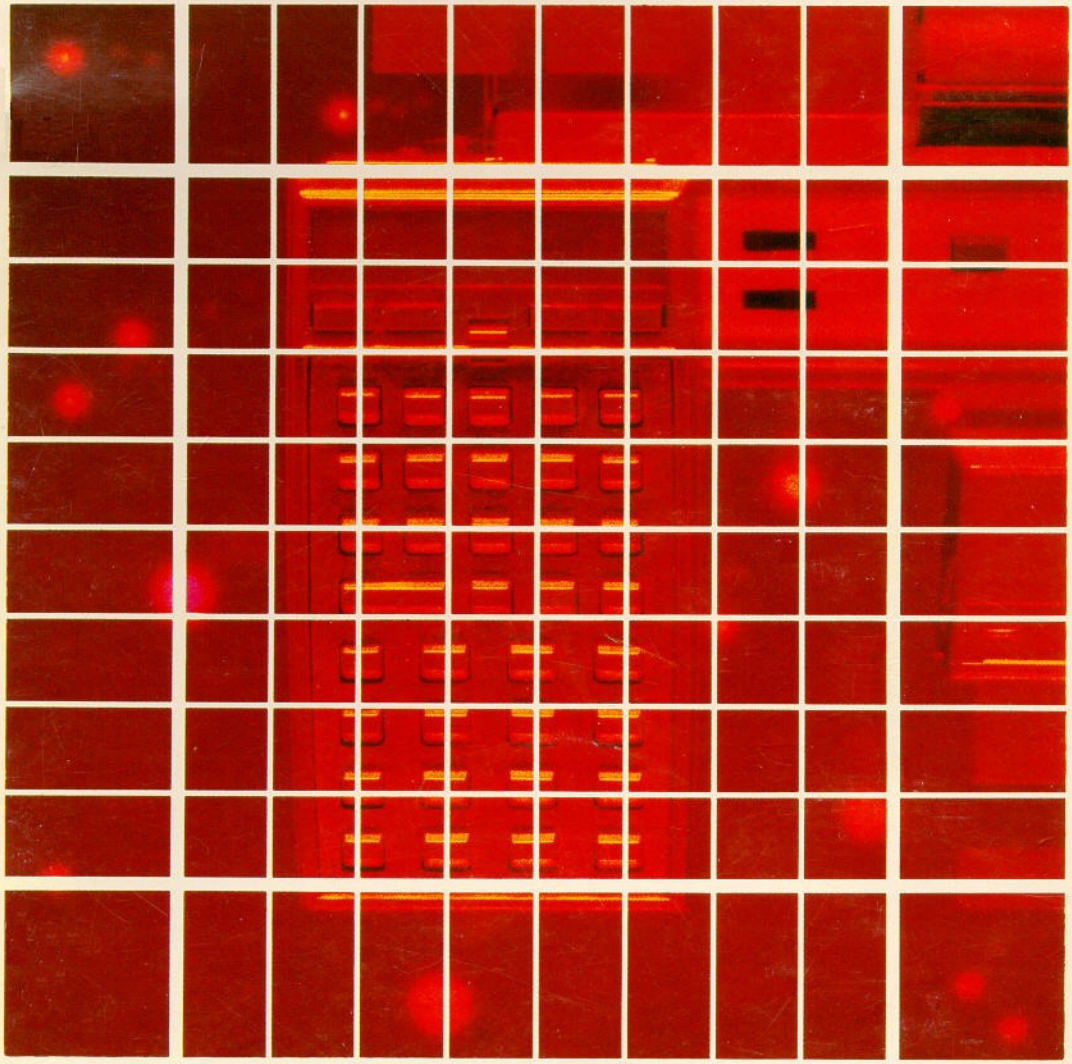
HEWLETT-PACKARD

HP-41C

OPERATING MANUAL

A Guide for the Experienced User



HP-41C Operating Manual

A Guide for the Experienced User

June 1980

00041-90259

Printed in U.S.A.

© Hewlett-Packard Company 1980

Contents

Introduction 5

Section 1: HP-41C Fundamentals 6

Calculator Modes 6

[ON] 6

[USER] 6

[PRGM] 6

[ALPHA] 6

Status Annunciators 7

Preview and Null 7

Prompting and Parameter Specification 7

Single-Key Parameter Specification 8

Indirect Parameter Specification 8

Catalogs 12

Display Execution of Functions and Programs 13

Display Editing and Clearing 15

Storing and Recalling Data 15

Viewing Register Contents 16

Clearing Data Storage Registers 16

Section 2: Memory 17

ALPHA Register 18

Main Memory 18

Main Memory Allocation 18

Changing Main Memory Allocation 18

Expanding Main Memory 20

Clearing Main Memory 23

Program Memory 24

Section 3: Alpha Mode 26

ALPHA Register 26

Displaying the ALPHA Register 27

Clearing and Editing Alpha Strings 28

Clearing the ALPHA Register 28

Editing Alpha Strings 28

Shifting Strings in the ALPHA Register 29

Storing Strings From the ALPHA Register 29

Recalling Data Into the ALPHA Register 29

Section 4: User Mode 31
Assigning Functions and Programs to Keys 31
User Mode Operation 33
Assignments of Local Alpha Labels 33

Section 5: Programming Basics 35
Loading a Program 35
Running a Program 37
Program Components 37
Program Lines 37
Labels 38

Section 6: Program Editing 39
Positioning Within Program Memory 39
Positioning With [GTO] [•] 39
Positioning With [RTN] 39
Positioning With Catalog 1 39
Single Step and Back Step 40
Deleting and Correcting Program Instructions 41
Deleting Instructions 41
Inserting Instructions 42
Clearing Programs 42

Section 7: Program Interruptions 44
Using ***STOP*** and [R/S] 44
STOP 44
Keyboard Stops 44
Using ***PROMPT*** 44
Using ***PSE*** (Pause) 44
Using ***OFF*** 45
Error Stops 45

Section 8: Branching and Looping 46
Using [GTO] in a Program 46
Label Searches 46
Global Label Searches 46
Local Label Searches 47
Conditional Functions 48
Controlled Looping 49

Section 9: Subroutines 50
Subroutine Calls 50
Subroutine Limits 51

Section 10: Flags 52
Section 11: Peripheral Devices 55
Section 12: Error and Status Messages 57
Function Index 59
Index 69

Introduction

This *HP-41C Operating Manual* is intended to be *A Guide for the Experienced User*. It supplements, but does not replace, the *HP-41C Owner's Handbook and Programming Guide*. If you have considerable experience with other programmable calculators, you may want to make this manual your primary learning tool as you get acquainted with the new and unique features of the HP-41C. If you choose to work through the owner's handbook first, you'll find this manual a convenient reference to important HP-41C operating characteristics. For the most part, this manual covers much of the same material as the owner's handbook, but in a somewhat abbreviated, reference-oriented style. The description of certain basics such as RPN and mathematical functions has not been repeated in this manual.

As you may already have discovered, functions and programs can be executed not only from the keyboard, but also "from the display." For clarity, functions that do not appear printed on or above a key (on the Normal mode keyboard or on the Alpha keyboard) are *always* shown throughout this manual as they would appear in the display, like ***DSE*** (unlike the owner's handbook, which shows all functions in "key boxes"). These functions can be executed *only* from the display (unless you assign them to a key).

Section 1

HP-41C Fundamentals

Calculator Modes

The four calculator modes, Normal, Alpha, User, and Program, are selected by the mode keys located just below the display window. The calculator is in Normal mode if it is *not* in Alpha, User, or Program mode.

ON

The ON key turns the calculator on and off. After about 10 minutes of inactivity, the calculator turns itself off automatically to prolong battery life. (Continuous Memory maintains programs, data, and calculator status even while the calculator is off.)

USER

The USER mode key switches the calculator into and out of User mode. In User mode, the user's "customized" keyboard is active. The **USER** annunciator is lit in User mode. Whenever the calculator is in Alpha mode, User mode is deactivated (though the **USER** annunciator remains lit).

PRGM

The PRGM mode key switches the calculator into and out of Program mode. In Program mode, keys pressed are not executed but instead are stored as program steps for later execution. The **PRGM** annunciator is lit in Program mode. When the **ALPHA** annunciator is lit, pressing the PRGM key also switches the calculator out of Alpha mode and turns the **ALPHA** annunciator off.

ALPHA

The ALPHA mode key switches the HP-41C into and out of Alpha mode. In Alpha mode, the Alpha keyboard, with its alphabetic, numeric, and special characters, becomes active. Part of the Alpha character set is printed in blue on the lower faces of the function keys; the entire character set is shown in the Alpha keyboard on the back of the calculator. Characters and functions shown above keys on the Alpha keyboard are accessed by pressing the ■ shift key first. The **ALPHA** annunciator is lit in Alpha mode.

Status Annunciators

Status annunciators appear along the bottom of the display. In addition to the **USER**, **PRGM**, and **ALPHA** annunciators mentioned above, the following annunciators may be displayed.

BAT indicates that the batteries are low; about 5 to 15 days of calculator operating time remain (using alkaline batteries) after **BAT** first appears. For more information about battery life, refer to page 240 of the owner's handbook.

GRAD or **RAD** indicates that the calculator is in Grads or Radians mode for trigonometric and rectangular/polar functions. If neither **GRAD** nor **RAD** appears, the calculator is in Degrees mode.

SHIFT indicates that the gold shift key (■) has been pressed.

0 1 2 3 4 indicates that the corresponding flag—0, 1, 2, 3, or 4—is set (true).

Preview and Null

Function Preview. When an HP-41C function key is pressed, the name of the function to be executed appears briefly in the display; when the key is released, that function is executed. This preview is particularly helpful in User mode, when the function executed by a key may not be the one indicated on the keyboard.

Function Null. If a function key is held down for longer than about a half-second, the function is nulled; that is, the function is not executed and the word ***NULL*** appears in the display. The ■ key can be nulled by pressing it again; this turns off the **SHIFT** annunciator.

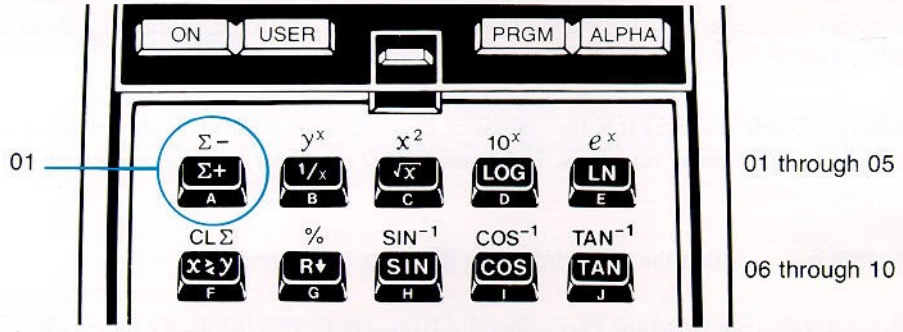
Prompting and Parameter Specification

The presence of one or more prompt signs (_) in the display indicates that the calculator expects input. If digit entry or Alpha entry has not been terminated, the display will show a single prompt sign; this signifies that new digits or Alpha characters keyed into the display will be appended to the number or Alpha string already there.

After certain functions have been specified for execution—such as [FIX] , [STO] , or [GTO] [•]—the display will show the function name followed by one, two, or three prompt signs. This generally indicates that a parameter—such as a label or an address—of one, two, or three digits (respectively) is expected.

Single-Key Parameter Specification

Unless the calculator is in Alpha mode, a one- or two-digit parameter between 0 and 10 can be entered by pressing the appropriate key on one of the two top rows:



[Σ+] through [LN] correspond to 01 through 05, respectively; [x≷y] through [TAN] correspond to 06 through 10, respectively. If a one-digit parameter is expected, only the rightmost digit (1 through 5 or 6 through 0) is used.

Indirect Parameter Specification

The parameters for most HP-41C functions can be specified indirectly as well as directly. With indirect parameter specification, instead of entering the parameter itself in response to the function prompt, you enter the address of a register (the "indirect register") that contains the parameter. This feature is particularly useful when the function is executed in a program and the value of the parameter depends on previous calculations in the program. In addition, accessing any of the extended registers $R_{(100)}$ through $R_{(318)}$ can only be done indirectly.

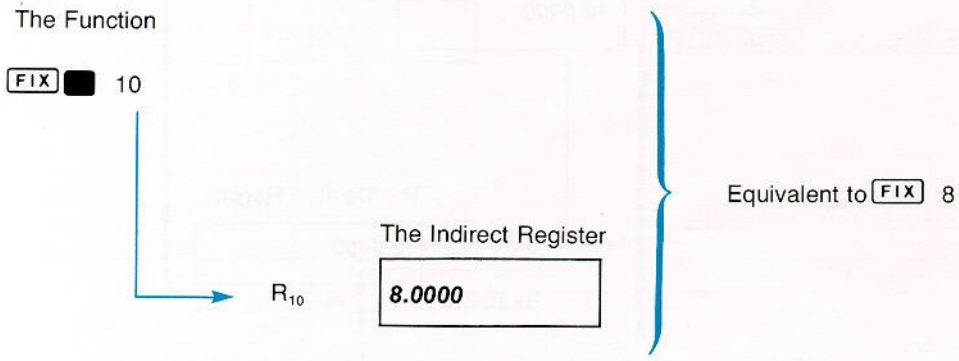
To specify a parameter indirectly:

1. Store the value of the parameter in the indirect register. This register can be any storage register R_{00} through R_{99}, any stack register, or the LAST X register. The calculator takes the parameter value to be the integer portion of the absolute value of the register contents.
2. In response to the function prompt, press [■]. The display at this point will show ***IND*** __ after the function name.
3. Specify the indirect register. If it is:
   - R_{00} through R_{99}, key in the two-digit address (i.e., the number of the register).
   - A stack register or the LAST X register, key in [•] followed by X, Y, Z, T, or L. You need not press the [ALPHA] key before these letter keys.

For example, to execute [FIX] 8 indirectly, using R_{10} as the indirect register:

Keystrokes	Display	
8 [STO] 10	***8.0000***	Stores the parameter 8 in register R_{10}. (Display shown assumes FIX 4 is in effect.)
[■] [FIX] [■]	***FIX IND __***	Pressing [■] after the function key specifies that the two-digit number entered next is the address of a register that contains the parameter, rather than the parameter itself.
10	***8.00000000***	Specifies the indirect register. FIX 8 is now in effect.

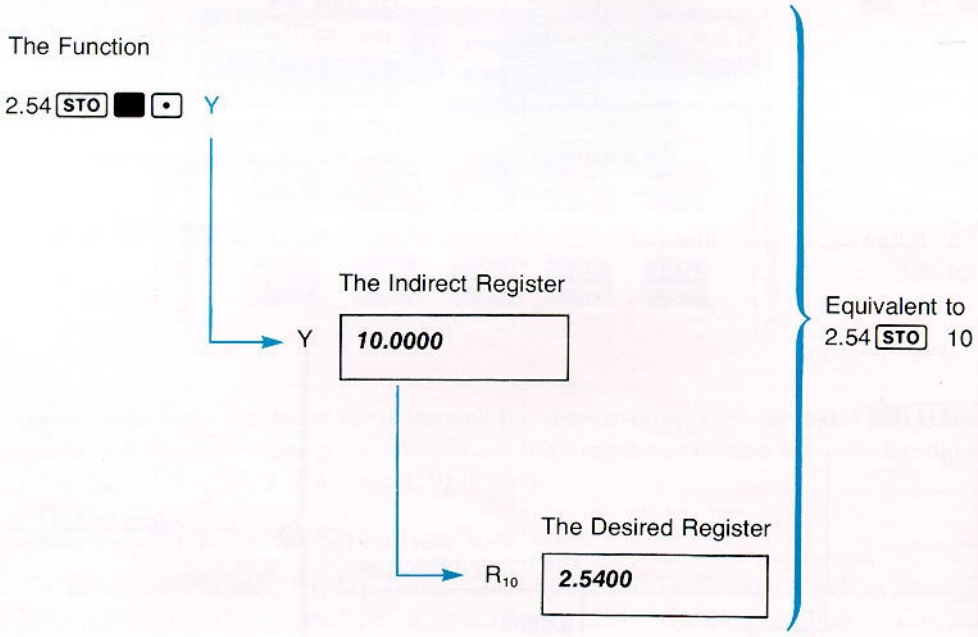
The following diagram illustrates what happened when we specified FIX 8 indirectly:



For most of the functions whose parameters can be specified indirectly, the parameter is a register address. For example, to store the number 2.54 into register R_{10}, using the Y-register as the indirect register:

Keystrokes	Display	
10 [ENTER↑]	***10.0000***	Stores the parameter 10 in Y-register. (Display shown assumes FIX 4 is in effect.)
2.54 [STO] [■] [•] Y	***2.5400***	The number 2.54 is now stored in R_{10}. (The character Y is entered by pressing the [×] multiplication key, which has Y printed on its lower face.)

The following diagram illustrates what happened when we stored 2.54 into R_{10} indirectly:

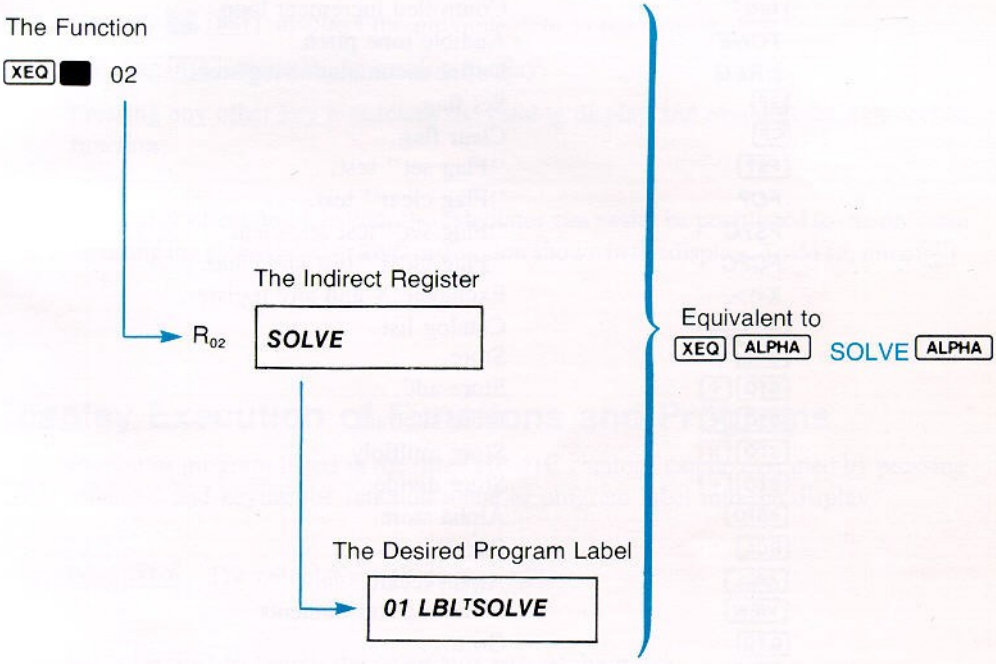


For GTO and XEQ, the parameter specified is either a global Alpha label or a numeric label. For example, to execute a program named SOLVE using R_{02} as the indirect register:

Keystrokes

ALPHA SOLVE ASTO 02 ALPHA	Stores the parameter SOLVE in register R_{02}.
XEQ ■ 02	Executes the program beginning with label SOLVE.

The following diagram illustrates what happened when we executed SOLVE indirectly:



With indirect parameter specification, the function is executed after the second key of the indirect register address is entered. If the register specified as the indirect register is not currently allocated to data storage, the calculator will display ***NONEXISTENT***.

Parameters for the following HP-41C functions can be specified indirectly:

FIX	FIX display format.
SCI	SCI display format.
ENG	ENG display format.
DSE	Controlled decrement loop.
ISG	Controlled increment loop.
TONE	Audible tone pitch.
ΣREG	Define accumulation registers.
SF	Set flag.
CF	Clear flag.
FS?	"Flag set" test.
FC?	"Flag clear" test.
FS?C	"Flag set" test and clear.
FC?C	"Flag clear" test and clear.
X<>	Exchange X and any register.
CATALOG	Catalog list.
STO	Store.
STO +	Store add.
STO −	Store subtract.
STO ×	Store multiply.
STO ÷	Store divide.
ASTO	Alpha store.
RCL	Recall.
ARCL	Alpha recall.
VIEW	View register contents.
GTO	Go to.
XEQ	Execute.

Catalogs

Three catalogs of functions and programs are maintained in the HP-41C:

Catalog 1: User-Written Program Catalog. A list of all global labels and ***END*** instructions in user-written programs. Programs are displayed in the order they are stored in program memory; within each program, the global labels and the ***END*** instruction are displayed beginning with the lowest line number.

Catalog 2: Extension Catalog. A list of all functions and programs available on calculator extensions or peripheral devices (such as the card reader, printer, or application modules) that are currently connected to the HP-41C. The functions in this list are grouped by extension.

Catalog 3: Standard Function Catalog. A list of all standard functions, including both those shown on the Normal mode keyboard and those that are "hidden." This listing is alphabetical.

Pressing [■] [CATALOG] followed by 1, 2, or 3 causes the display to cycle through the catalog designated. During a catalog display, pressing any key other than [ON] or [R/S] will slow the listing, and pressing [R/S] will halt the listing.

With the display of any catalog halted:

- Pressing [SST] displays the next item in the catalog.
- Pressing [■] [BST] displays the previous item in the catalog.
- Pressing [R/S] resumes the catalog display.
- Pressing any other key terminates the catalog display and executes the appropriate function.

With the display of catalog 1 halted, the calculator can easily be positioned to the program line containing the global label or ***END*** instruction shown in the display. To do so, press [←] .

Display Execution of Functions and Programs

Any function or program listed in the three HP-41C catalogs can be executed by pressing [XEQ] *(execute)* and keying the function name or program label into the display:

1. Press [XEQ] . The calculator will display ***XEQ*** __.

2. Press [ALPHA] to switch the calculator into Alpha mode.

3. Key the function name or program label into the display by pressing the appropriate keys as identified on the Alpha keyboard. This keyboard is shown inside the back cover of this manual and on the back case of the calculator. As with functions on the Normal mode keyboard, characters shown above the keys on the Alpha keyboard are accessed by pressing the [■] shift key first.

4. Press [ALPHA] to switch the calculator out of Alpha mode. If a function being executed does not require a parameter—or if a program is being executed—the function or program is executed when the calculator leaves Alpha mode.

5. If a function being executed requires a parameter—as does ***TONE***, ***FC?***, or ***SIZE***—when the calculator leaves Alpha mode, ***XEQ*** will disappear from the display. The display will then show the function name followed by the appropriate number of prompt signs. The function will be executed when the parameter is keyed in.

For example, to evaluate the expression $e^x - 1$ when $x = 0.45$, key .45 into the displayed X-register, then execute the function ***E↑X−1*** as follows:

Keystrokes	Display	
.45	***.45_***	Key the value of x into the display.
[XEQ]	***XEQ __***	Press [XEQ]. The calculator responds by prompting: *Execute what?*
[ALPHA]	***XEQ _***	Switch the calculator into Alpha mode.
E	***XEQ E_***	Press the E key on the Alpha keyboard.
↑ ([■] N)	***XEQ E↑ _***	↑ is a shifted character of the N key on the Alpha keyboard.
X	***XEQ E↑X_***	Press the X key on the Alpha keyboard. This should not be confused with the [×] (multiplication) key on the Normal mode keyboard.
− ([■] Q)	***XEQ E↑X− _***	− is a shifted character of the Q key on the Alpha keyboard.
1 ([■] Z)	***XEQ E↑X−1_***	The *character* 1 is a shifted character of the Z key on the Alpha keyboard. This is *not* the same as the *digit* 1 on the Normal mode keyboard.
[ALPHA]	***0.5683***	Switch the calculator out of Alpha mode, whereupon the function is executed.

As another example, to execute the function ***FS?C*** on flag 9:

Keystrokes	Display	
[XEQ]	***XEQ __***	
[ALPHA]	***XEQ _***	
FS?C	***XEQ FS?C _***	Key the function name into the display.
[ALPHA]	***FS?C __***	Switch the calculator out of Alpha mode. The calculator prompts for a two-digit parameter—in this case, a flag number.
09	***NO***	The function is executed when the two digits have been keyed in. (The calculator's answer indicates that flag 9 was not set.)

If the function name or program label is not currently listed in one of the three catalogs, the calculator will display ***NONEXISTENT***.

Many functions provided by the HP-41C can be executed *only* by pressing [XEQ] and keying in the function name, since there is no corresponding key on the Normal mode keyboard. (However, all such functions can be assigned to a key and then executed simply by pressing that key when the calculator is in User mode.) For many functions that can be executed both from the display and from the keyboard—such as ***SIN***—the name for display execution is identical to the name shown on the key. For others—such as ***SQRT*** *(square root)*—the name for display execution is different from that shown on the key. For each function listed in catalog 3, the name for display execution and the corresponding key(s) (if any) are listed in the Function Index at the back of this manual.

Display Editing and Clearing

If a prompt sign appears in the display (signifying that digit entry or Alpha entry has not been terminated), pressing [←] *(correction)* deletes the right-most digit from a number being entered or the right-most character from an Alpha string being entered.

In Normal mode, pressing [■] [CLX] *(clear x)* clears the entire displayed X-register, replacing its contents with zero. In Alpha mode, pressing [■] [CLA] *(clear Alpha)* clears (i.e., erases) the entire ALPHA register, including the portion not shown in the display.

If no prompt sign appears in the display (signifying that digit entry or Alpha entry has been terminated), pressing [←] has the same effect as pressing [■] [CLX] or [■] [CLA]. If the calculator is not in Alpha mode and only one digit appears in the display, pressing [←] has the same effect as pressing [■] [CLX]. Likewise, if the calculator is in Alpha mode and only the prompt sign appears in the display, pressing [←] has the same effect as pressing [■] [CLA].

If the display is currently showing the contents of a register because a ***VIEW*** or ***AVIEW*** instruction has been executed, pressing [←] will return the display to the X-register or (if the calculator is in Alpha mode) the ALPHA register. This can be accomplished in a program ([←] is not programmable) by using the ***CLD*** *(clear display)* instruction.

Pressing [←] also clears error messages from the display.

Storing and Recalling Data

To store data contained in the displayed X-register into any of storage registers R_{00} through R_{99}, into any of the stack registers, or into the LAST X register:

1. Press [STO]. The calculator will prompt for a two-character address with ***STO*** __.
2. Press the digit keys for the desired storage register address (00 through 99); or [•] followed by X, Y, Z, T, or L (you need not press the [ALPHA] key first) to select the desired stack register or the LAST X register.

When data, whether a number or an Alpha string, is stored into a register, that data writes over any data previously contained in the register. Each storage register can hold either a number (as a 10-digit mantissa plus a 2-digit exponent of 10) or an Alpha string (of up to six characters).

To recall data (either a number or an Alpha string) from a register into the X-register, press the [RCL] key and respond to the prompt in the same manner as when storing data. Recalling data from a register is non-destructive—that is, the data is merely copied into the X-register; it also remains in the storage register. If the stack lift is enabled, recalling data into the X-register causes the stack to lift first.

Viewing Register Contents

To see the contents of any storage register, stack register, or the LAST X register—*without* recalling it into the X-register:

1. Press [■] [VIEW] .
2. Specify, directly or indirectly, the register desired—just as with [STO] or [RCL].

Unlike [RCL] , however, [VIEW] does not copy the contents of the specified register into the X-register—it merely displays it. Neither the specified register nor the X-register is altered.

Pressing [←] terminates the viewing and returns the contents of the X-register to the display. Pressing any other key terminates the viewing and also performs that key's function upon the contents of the X-register—just as if the display had been showing the contents of the X-register before that key was pressed.

In a program, the viewing can be terminated by executing ***CLD*** *(clear display)*. This returns the }- label execution indicator to the display.

Clearing Data Storage Registers

To clear a single data storage register, store zero into it.

To clear all data storage registers at once, execute the ***CLRG*** *(clear registers)* function.

To clear all storage registers at once (as well as all other information maintained in the calculator's Continuous Memory):

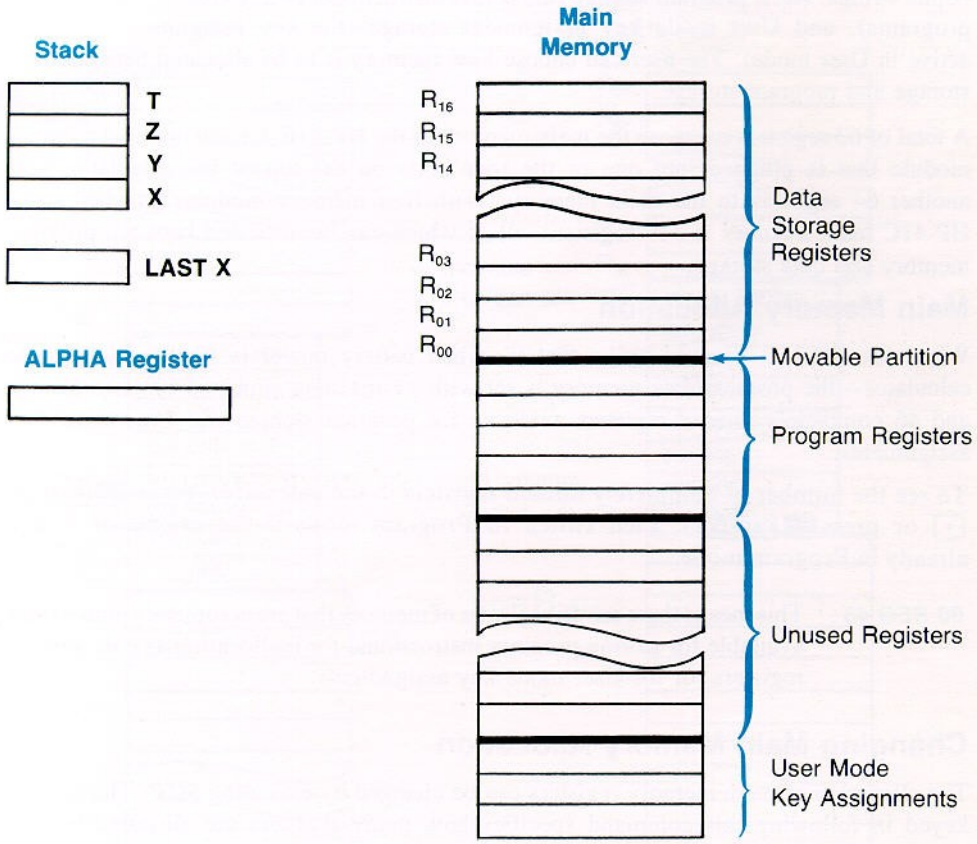
1. Turn the calculator off.
2. Hold down the [←] key and press [ON] .
3. Release the [←] key.

You need not clear a storage register before storing data into it; the storing operation automatically clears the register before the data is stored.

Section 2

Memory

The Continuous Memory of the HP-41C maintains its contents—including that of plugged-in memory modules—even while the calculator is turned off. Certain flags (refer to section 10) are also maintained.



HP-41C Memory

ALPHA Register

The ALPHA register is a separate register that holds Alpha characters, which are the characters shown on the Alpha keyboard. An Alpha string (i.e., a series of characters) in the ALPHA register can include up to 24 of these characters, with each period, comma, colon, and space counting for one of the 24.

While the calculator is in Alpha mode, the display usually shows a portion of the contents of the ALPHA register. (Refer to page 26 for the exceptions.) Only 12 of the characters in the ALPHA register can be displayed at one time; periods, commas, and colons, since they fit between other characters, do *not* count for one of these 12.

Main Memory

The main memory of the HP-41C is used for data storage (numbers, results, constants, Alpha strings, etc.), program storage (the actual instructions, or keystrokes, that make up programs), and User mode key assignment storage (the key assignments that are active in User mode). The user can choose how memory is to be allocated between data storage and program storage.

A total of 63 registers compose the main memory of the HP-41C.* Each optional memory module that is plugged into one of the four ports on the top of the calculator adds another 64 registers to the main memory. With four memory modules attached, total HP-41C main memory is 319 registers, all of which can be allocated between program memory and data storage.

Main Memory Allocation

When the HP-41C "wakes up"—that is, when battery power is first applied to the calculator—the program/data memory is set with 17 registers allocated for data storage and 46 completely unused registers available for program storage and User mode key assignments.

To see the number of completely unused registers in the calculator, press ■ GTO . . or press ■ CATALOG 1, then switch to Program mode if the calculator is not already in Program mode.

00 REG 46 This means there are 46 *registers* of memory that are completely unused and available for storing program instructions, for reallocation as data storage registers, or for User mode key assignments.

Changing Main Memory Allocation

The allocation of main memory registers can be changed by executing ***SIZE***. The number keyed in following this command specifies how many registers are allocated to *data storage*. The remaining registers are automatically allocated to program storage and User mode key assignments.

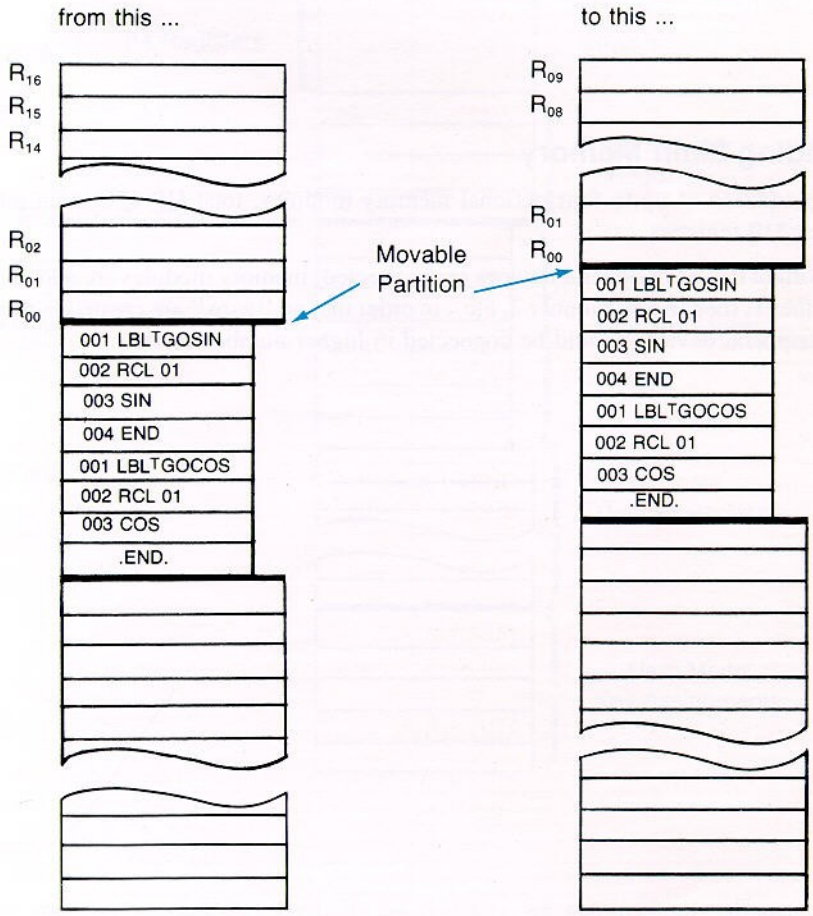
*There are actually 64 registers, but a portion of one register is utilized for a permanent ***END*** at the end of program memory.

To change main memory allocation:

1. Execute ***SIZE***. (***SIZE*** is not programmable.) ***SIZE*** ___ will appear in the display.
2. Key in the number of registers to be allocated to data storage: a three-digit number from 000 through 319.

When ***SIZE*** is executed, both data and program information—as well as the memory partition separating them—are moved up or down in program memory.

If ***SIZE*** specifies a smaller number of data storage registers than the number currently allocated to data storage, the information and the partition move up. Any data in the highest-numbered storage registers will be lost off the top of memory. For example, if main memory were allocated as on the left, executing ***SIZE 010*** would change the allocation



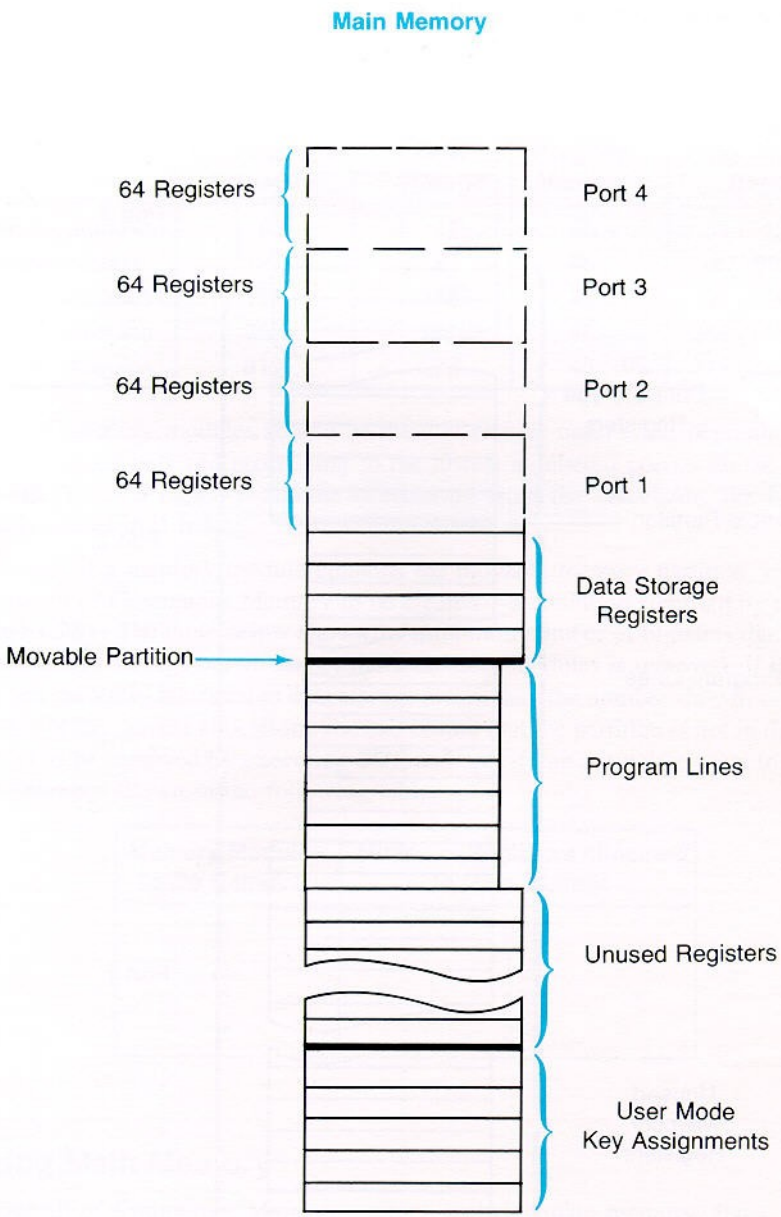
If ***SIZE*** specifies a larger number of data storage registers than the number currently allocated to data storage, the partition and the data and program information move down. If the calculator displays ***PACKING*** followed by ***TRY AGAIN***, there may be insufficient unused registers available in memory to accommodate the added data storage registers. If the same result is obtained following another execution of ***SIZE*** (specifying the same number of data storage registers), the allocation specified will not be possible unless program instructions or key assignments (of functions and programs listed in catalogs 2 and 3) are deleted from memory or an additional memory module is added.

To determine how many completely unused registers remain available for program storage after executing ***SIZE***, press ■ GTO • •, then set the calculator temporarily to Program mode. The display will show ***00 REG*** **nn**, where **nn** is the number of completely unused registers available for program storage. If ***SIZE*** is executed while an executing subroutine is halted, the calculator loses all pending ***RTN*** and ***END*** instructions.

Expanding Main Memory

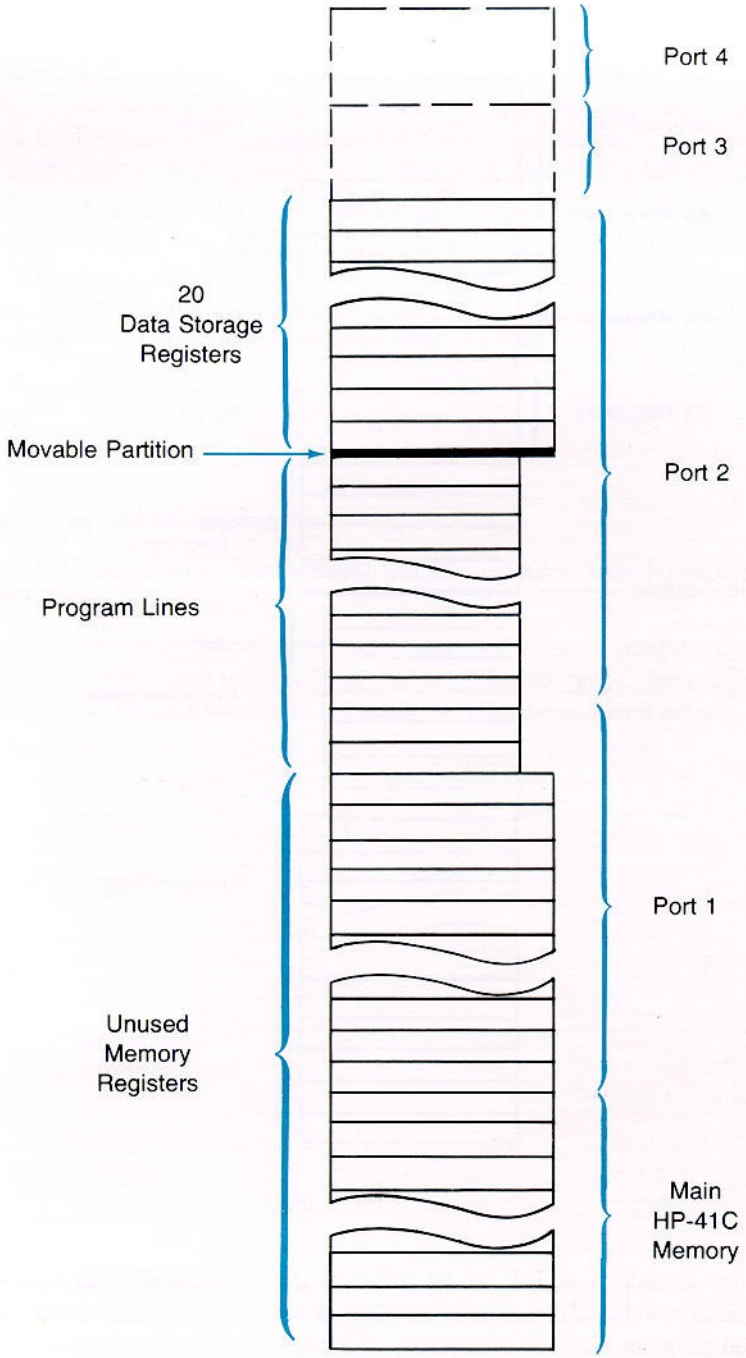
With the addition of up to four optional memory modules, total HP-41C main memory can reach 319 registers.

It is important that as peripheral devices are connected, memory modules are added first in port number 1, then in port number 2, etc., in order that no "gaps" are created in memory. Other peripheral devices should be connected in higher-numbered ports.



As each memory module is added, its 64 registers are automatically allocated in the data storage area of memory. However, using ***SIZE***, registers may be reallocated between data storage and program memory in any way desired.

For example, with two memory modules plugged into the HP-41C, executing ***SIZE 020*** would reallocate the registers as shown below.



SIZE 020 Allocation With Memory Modules in Ports 1 and 2

The following table shows the initial allocation and the maximum number of registers that can be allocated to data storage or to program memory.

Calculator Plus	Maximum Data Storage Registers	Initial Register Allocation		Maximum Program Memory Registers
		Data Storage	Program Memory	
No Memory Modules	63	17	46	63 (445 bytes)
1 Memory Module	127	81	46	127 (893 bytes)
2 Memory Modules	191	145	46	191 (1341 bytes)
3 Memory Modules	255	209	46	255 (1789 bytes)
4 Memory Modules	319	273	46	319 (2237 bytes)

Removal of memory modules should be done in reverse order (i.e., beginning with the highest-numbered port and proceeding to the lowest-numbered port) with the calculator turned off. When a memory module is removed from the calculator, the information previously stored in it is lost.

Furthermore, if a memory module contains the movable memory partition, removing it will cause all of Continuous Memory to be cleared—a condition signified by the display ***MEMORY LOST***. The table below shows the minimum number of registers that should be allocated to data storage before one or more memory modules is removed. If the number of registers currently allocated to data storage is less than the number shown, or if you are not certain of the current allocation, you can ensure that the partition is not in the memory module(s) to be removed by executing ***SIZE*** and specifying a number equal to or greater than the number shown in the following table.

Memory Modules To Be Removed	Minimum Registers Allocated to Data Storage
1	064
2	128
3	192
4	256

Clearing Main Memory

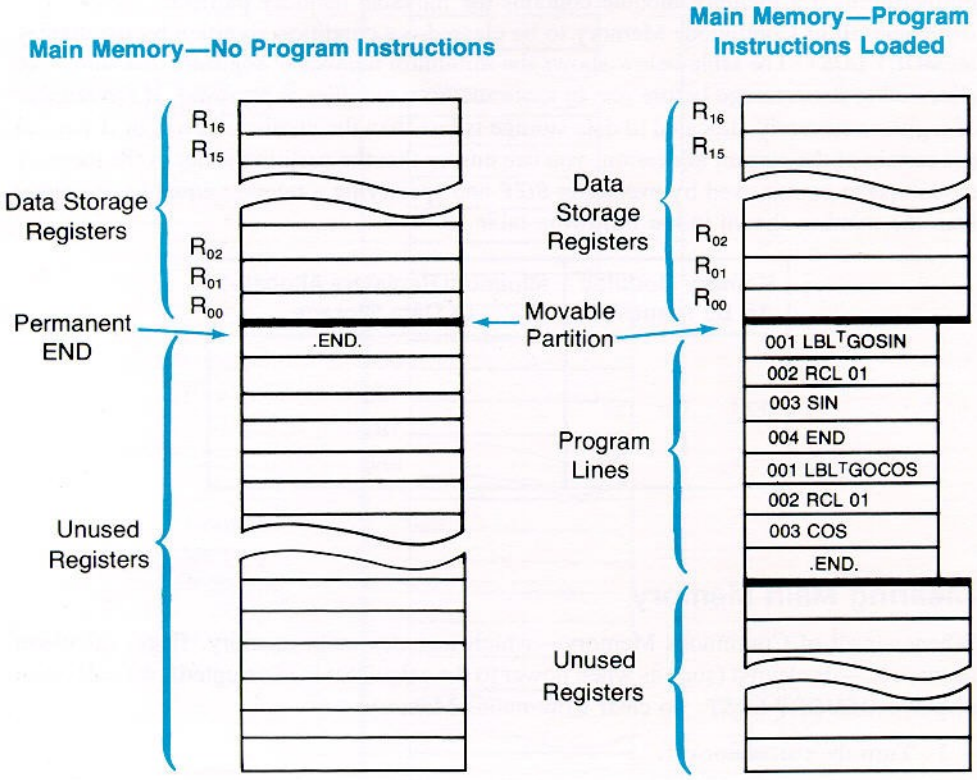
Whenever all of Continuous Memory—which includes main memory, flags, calculator status, etc.—is cleared (such as when power to the calculator is interrupted), the calculator displays ***MEMORY LOST***. To clear Continuous Memory:

1. Turn the calculator off.
2. Hold down the ← key and press ON .
3. Release the ← key.

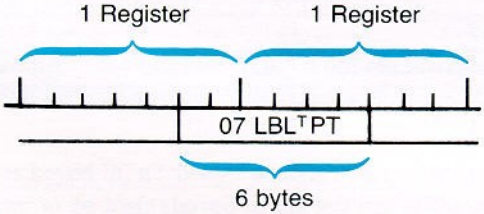
Program Memory

As instructions are keyed into a program, they are stored in unused memory registers. Each register consists of seven bytes. Functions may occupy one, two, three, or four bytes, depending on the particular function; the number of bytes occupied by each function is listed in the Function Index at the back of this manual. Numeric data occupies one byte for each digit in the number, plus another byte for each [•], [CHS], and [EEX] keyed in with the data. Alpha data occupies one byte for each character in the Alpha string, plus one additional byte for the entire string.

Each function, number, or Alpha string in a program is considered to be a separate line of program memory. These program lines are numbered consecutively, beginning with 1, within each program. The number of program lines depends on how many functions, numbers, and Alpha strings are in the program; while the number of registers occupied by these program lines depends upon the particular functions and the lengths of the numbers and Alpha strings.



The bytes occupied by the instruction in a program line may not all be contained in the same register. For example, in the following illustration the instruction ***LBL ᵀ PT*** occupies the last two bytes in one register and the first four bytes in the next register. The remaining bytes in these registers would be used for the instructions in the program lines immediately preceding and following the ***LBL ᵀ PT*** instruction.



In certain conditions, some of the bytes in a register may be unused. This happens when other instructions using bytes in that register are deleted and when a number is keyed into a program. It may also happen when a function, number, or Alpha string is inserted either within a program, or at the end of a program that is not the last program in memory (i.e., whenever an instruction is inserted anywhere but at the end of program memory). If there were not already unused bytes available between the bytes containing the instructions in the adjacent program lines when a new instruction is inserted, all subsequent instructions are moved seven bytes downward in memory, the new instruction is placed into the appropriate number of bytes, and the rest of the bytes in that register remain unused.

Any unused bytes interspersed in program storage registers can be utilized by executing ***PACK***. When program memory is packed, instructions within all programs are moved up into unused bytes, and—depending on the number of those bytes—new registers may be made available for program storage. Program memory is also automatically packed whenever [GTO] [•] [•] is executed, when complete programs are deleted (by executing ***CLP***), and when an instruction is inserted or added for which there are not sufficient unused bytes currently available in memory.

Section 3

Alpha Mode

Pressing the [ALPHA] key switches the HP-41C into and out of Alpha Mode. The calculator is also switched into Alpha mode when ***AON*** is executed and switched out of Alpha mode when ***AOFF*** is executed.

While the HP-41C is in Alpha mode:

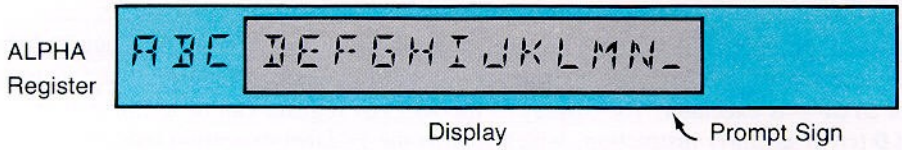
- The Alpha keyboard is active. This keyboard is shown inside the back cover of this manual and on the back case of the calculator. Characters and functions shown above the keys on the Alpha keyboard are accessed by pressing the ■ shift key first.
- The display shows the ALPHA register (instead of the X-register), and characters keyed into the display are automatically entered into the ALPHA register except in either of the following conditions:
  - The display shows ***ASN***, ***CLP***, ***GTO***, ***LBL***, or ***XEQ***, followed by one or more prompt signs. If so, up to seven characters can be entered as the argument of the operation.
  - The calculator is in Program mode.

If the calculator is in Program mode as well as Alpha mode, characters keyed in (up to 15) are entered as data into a program line (unless the display shows one of the five instructions listed above). The display of the program line will show a ^{T} signifying that the characters following comprise an Alpha string. When that line is executed, the ALPHA register is cleared and the data is entered into the ALPHA register (but *not* automatically into the display).

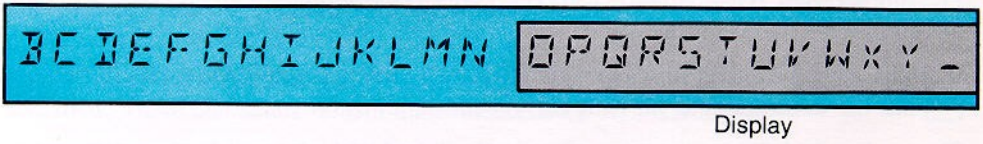
ALPHA Register

The ALPHA register can hold up to 24 Alpha characters. Each period, comma, and space counts for one of these 24. A display of the ALPHA register can show only 12 of these characters at a time (and often only 11 plus the prompt sign); but periods, commas, and colons, since they fit between other characters, do *not* count for one of these 12.

To enter Alpha characters into the displayed ALPHA register, switch the calculator to Alpha mode, then press the appropriate keys as shown on the Alpha keyboard. Characters are written in the display from left to right as keys are pressed. If you continue to enter characters after 11 characters have been keyed in, the characters in the display shift to the left. Characters shifted out of the display remain in the ALPHA register.



As the 24th character is keyed in, a "beep" sounds to warn that the next character keyed in will cause a character to be lost, shoved to the left out of the ALPHA register. When a full 24 characters have been placed into the ALPHA register, you may continue to add characters at the right of the display; but for each character added, the calculator will again "beep" and one character will be lost at the left of the ALPHA register.



The prompt sign (_) indicates that as Alpha keys are pressed, the corresponding characters will be appended to the existing Alpha string. The prompt sign disappears when Alpha entry is terminated.

The following operations terminate Alpha entry:

[ALPHA] [ASTO] [AVIEW] [BST] [CLA] ***OFF*** [ON] [R/S] [SST]

Displaying the ALPHA Register

Whenever the calculator is switched to Alpha mode (unless it is in Program mode or a running program executes ***AON***), the 12 left-most characters in the ALPHA register appear in the display, then the remaining characters in the ALPHA register (if any) are "scrolled" through the display.

To display the contents of the ALPHA register in a program or without switching to ALPHA mode, execute ***AVIEW*** *(Alpha view)*. When the calculator is already set to Alpha mode, pressing ■ [AVIEW] terminates Alpha entry before displaying the contents of the ALPHA register.

When the calculator is not in Alpha mode, pressing ← terminates the display of the ALPHA register and returns the contents of the X-register to the display—just as it does with ***VIEW***. Likewise, pressing any other key terminates the ALPHA register display and also executes that key's function.

When the calculator is set to Alpha mode, of course, pressing ← after ***AVIEW*** clears the ALPHA register.

In a running program, the ALPHA register is displayed when ***AVIEW*** is executed. It is not displayed when ***AON*** is executed unless and until a program interruption—such as ***PSE*** or ***STOP***—is executed. The display of the ALPHA register can be terminated using the ***CLD*** *(clear display)* instruction, which returns the ⊦- label execution indicator to the display.

Clearing and Editing Alpha Strings

Clearing the ALPHA Register

Pressing ■ CLA *(clear Alpha)* clears (i.e., erases) the entire ALPHA register, including the portion not shown in the display.

If Alpha entry has been terminated (signified by no prompt sign in the display), pressing ← has the same effect as pressing ■ CLA .

The displayed ALPHA register is automatically cleared (if Alpha entry has been terminated) when a new character is keyed into it from the keyboard or when a program executes an instruction consisting of only an Alpha string.

Editing Alpha Strings

If Alpha entry has not been terminated (signified by the prompt sign appearing in the display):

- Characters are entered into the display when the corresponding keys are pressed.
- Pressing ← deletes the right-most character from the display.

To delete or append characters in the ALPHA register after Alpha entry has been terminated, first reactivate Alpha entry by pressing ■ APPEND .* This restores the prompt sign to the display.

If a program is to append characters to those already in the ALPHA register, with the calculator in Program mode press ■ APPEND , then key the characters into the same program line. The display of that program line will show ⊦- after the ᵀ, signifying that the Alpha string is to be appended to that already in the ALPHA register. If the ⊦- does not appear before the Alpha string (i.e., APPEND has not been keyed into that program line), the characters in that line will replace the existing string in the ALPHA register rather than be appended to it.

---

*APPEND is a shifted function on the Alpha keyboard. To execute it, therefore, set the calculator to Alpha mode and press ■ K.

Shifting Strings in the ALPHA Register

Executing ***ASHF*** *(Alpha shift)* shifts each character in the ALPHA register six places to the left. The six characters previously at the left are lost, and the next six characters are then positioned at the left of the ALPHA register.

Storing Strings From the ALPHA Register

Alpha strings in the ALPHA register can be stored into any storage register, stack register, or the LAST X register using the ***ASTO*** *(Alpha store)* function.

To store the left-most six characters from the ALPHA register into a register:

1. Execute ***ASTO*** or (in Alpha mode) press ■ [ASTO] .
2. Specify, directly or indirectly, the register desired—just as with [STO] .

ASTO copies the left-most six characters from the ALPHA register into the register specified. These characters remain undisturbed in the ALPHA register.

If additional characters from the ALPHA register are to be stored, execute ***ASHF*** to position the next six characters at the left of the ALPHA register, then execute ***ASTO***. Executing ***ASHF*** and ***ASTO*** again stores the next six characters from the ALPHA register; doing so once more stores the last six characters. Since each register can hold only six characters, each ***ASTO*** should specify a different register.

Numeric characters (as well as alphabetic characters) stored from the ALPHA register using ***ASTO*** are considered to be Alpha data, not numeric data; therefore, they cannot be used for normal mathematical operations (such as storage register arithmetic). An attempt to do so will result in an ***ALPHA DATA*** error message. However, Alpha strings (alphabetic or numeric) can, after being stored in the X- and Y-registers, be compared using the [x=y?] and ***X≠Y?*** conditional functions.

Recalling Data Into the ALPHA Register

To recall data (either numeric or Alpha) from a register into the ALPHA register:

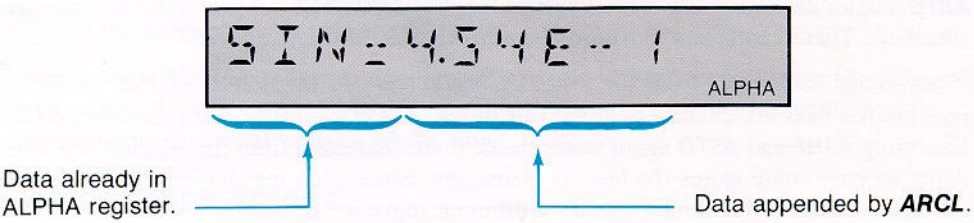
1. Execute ***ARCL*** or (in Alpha mode) press ■ [ARCL] .
2. Specify, directly or indirectly, the register desired—just as with [RCL] .

The data recalled is automatically appended at the right of the previous contents of the ALPHA register. If the resulting character string is longer than 24 characters, only the right-most 24 characters remain in the ALPHA register; the remaining characters are lost. The data in the register specified is not altered.

If the register specified after ***ARCL*** contains numeric data, the copy of that data in the ALPHA register is considered to be Alpha data. However, the characters appended in the ALPHA register are those determined by the display mode (FIX, SCI, or ENG) in effect. If that display mode is SCI or ENG, the exponent field is preceded by ***E*** and blank character positions preceding the exponent field are omitted. For example, the number 1.23456789 $\times$ 10^{44} would be recalled to the ALPHA register as follows:

Display Mode	Numeric Data as Displayed in X-Register	Corresponding Alpha Data in ALPHA Register
SCI 2	*1.23 44*	*1.23E44*
SCI 7	*1.2345679 44*	*1.2345679E44*
SCI 9	*1.2345678 44*	*1.234567890E44*

Numeric data—as well as Alpha data—that is recalled into the ALPHA register is appended to the data already in the ALPHA register. For example:



Unlike [RCL] terminating digit entry, ***ARCL*** does not terminate Alpha entry.

Section 4

User Mode

User mode gives you a "personalized" calculator at a press of the [USER] key. After programs (or subroutines), keyboard functions, "hidden" functions, or functions from attached peripheral devices have been assigned to HP-41C keys, switching to User mode activates this keyboard. Keyboard overlays and sticky-back labels can be used to identify User mode functions.

Assigning Functions and Programs to Keys

To assign a function or program to a key location:

1. Press ■ [ASN] in Normal or User mode. The calculator will display ***ASN*** _.

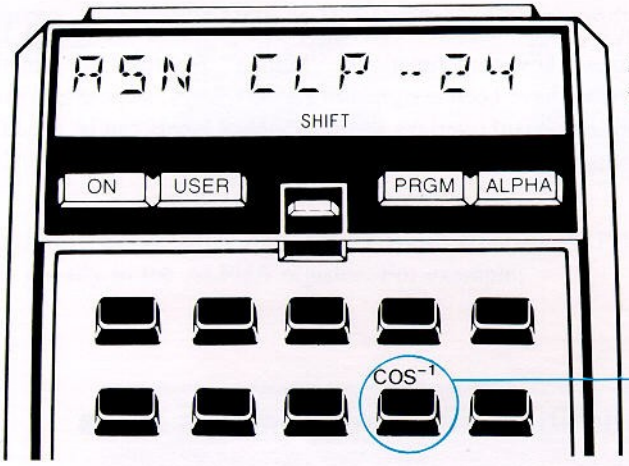
2. Press [ALPHA] to switch the calculator into Alpha mode.

3. Key in the function name or program label to be assigned. The program label to be assigned must be a global Alpha label, not a local Alpha label nor a numeric label. (The various kinds of labels are described on page 38.) Local Alpha labels are automatically assigned to certain keys, as discussed under Assignments of Local Alpha Labels, page 33.

4. Press [ALPHA] to switch the calculator out of Alpha mode.

5. Press the key (or ■ and the key) to which the function or program is to be assigned. If the key is held down, the display will show the function name or program label assigned, plus the keycode of the key location. If the key is held down until the display shows ***NULL***, the assignment is cancelled.

Keycodes are a row-column identification of a key location, reading from top to bottom and left to right. For example, the keycode for the [COS] key location is 24: that is, the second row of keys, fourth key in the row. Keycodes for shifted key locations are prefixed with a – (minus sign) in the display.



A total of 68 key locations are available for User mode key assignments. The only key locations to which functions and programs *cannot* be assigned are ■, [ON], [USER], [PRGM], and [ALPHA].

Any function name or program label that appears in any of the three HP-41C catalogs can be assigned to any key location (other than the five listed above) for User mode operation. An attempt to assign a function or program which is not currently listed in the calculator's catalogs will result in an error display of ***NONEXISTENT***.

To cancel a key assignment (i.e., reassign to a key location its original function), press ■ [ASN] [ALPHA] [ALPHA] followed by the key. The key's Normal mode function will then be active in User mode as well.

Examples:

■ [ASN] [ALPHA] STO [ALPHA] [1/x] — Assigns ***STO*** function to [1/x] key location for User mode.

■ [ASN] [ALPHA] CIRCLE [ALPHA] ■ [e^x] — Assigns program named ***CIRCLE*** to [e^x] key location for User mode. (Program must be in program memory.)

■ [ASN] [ALPHA] [ALPHA] [1/x] — Restores normal function of [1/x] in User mode.

User Mode Operation

Pressing the [USER] key takes the HP-41C into and out of User mode. In User mode, functions and programs assigned to keys become active. Thus, a single key on the keyboard can be used to execute four different functions, depending on whether the calculator is in User mode and whether the ■ shift key is pressed first.

In User mode, pressing a key to which a function name has been assigned is equivalent to pressing [XEQ] and specifying that function name. Likewise, when a key is pressed to which a program label has been assigned, the calculator begins execution at that label in program memory, just as if you had pressed [XEQ] and specified that program label.

If no function or program has been assigned to a particular key location, when that key is pressed—in User mode as well as in Normal mode—the function printed on (or above) the face of the key is executed (except as described below under Assignments of Local Alpha Labels).

To check whether a function or program has been assigned to a particular key location, with the calculator in User mode press the key and hold it down until the display shows ***NULL*** (so that the function or program will not be executed). The display will show the name of the function or program currently active at that key location. Names of programs are preceded by ᵀ.

Each User mode function or program assignment is maintained by Continuous Memory and remains assigned to that key until another function or program is assigned there, or until its Normal mode function is restored.

Assignments of functions or programs listed in catalogs 2 and 3 consume one register (seven bytes) for each odd-numbered assignment made. For example, the first assignment made consumes one register; the second assignment consumes no additional space; the third assignment consumes another full register; the fourth consumes no additional space; and so on. Assignment of a program listed in catalog 1 (i.e., user-written programs) does not consume any additional space in program memory beyond that used for storage of the program itself.

Assignments of Local Alpha Labels

When the calculator is in User mode, local Alpha labels are automatically assigned to each of the keys on the top two rows (unless a function name or another program label has been explicitly assigned to the key). The label assigned to each key corresponds to the Alpha character shown for the key on the Alpha keyboard: A through E on the top row, F through J on the second row, and a through e on the top row if the ■ key was pressed first. Because these labels are local (rather than global),

the calculator will begin execution at the specified label in program memory only if that label is found within the current program.* Thus, with the calculator in User mode, the function (or program) executed when one of these keys is pressed is determined according to the following priorities:

1. If a function or program has been assigned to that key, that function or program is executed.
2. If the label corresponding to the key pressed is within the current program, the calculator begins execution at that label.
3. If neither of the first two conditions is true, the function printed on (or above) the face of the key is executed. Execution of that function may take significantly more time in User mode than in Normal mode due to the time required for the label search. Therefore, to avoid waiting if you want to execute the key's Normal mode function, switch the calculator out of User mode before pressing the key.

If the label corresponding to one of these keys is within the current program, the preview seen in the display when the key is pressed shows ***XEQ*** followed by the label; otherwise, the preview shows the name of the Normal mode function of that key.

*Searches for local labels are conducted only within the program to which the calculator is currently set (i.e., between ***END*** instructions). (Refer to Local Label Searches, page 47, for more information.)

Section 5

Programming Basics

Loading a Program

To load a program into the calculator:

1. Switch the calculator to Program mode by pressing [PRGM].
2. Press [■] [GTO] [.] [.] to set the calculator to an unused portion of program memory. The display will show the number of completely unused memory registers available for new program instructions.
3. Press [■] [LBL] followed by a two-digit number or by the Alpha characters comprising the program name.
4. Key in functions, numbers, or Alpha strings, just as in Normal mode. In Program mode, however, these instructions are not executed, but they are remembered by the calculator in the order they are keyed in.

You may switch among the Normal, Alpha, and User mode keyboards, just as you do when you are using the HP-41C keyboard to solve problems, execute functions, or key in Alpha strings. All operations in the HP-41C, including "hidden" functions, Alpha mode operations, and reassigned User mode keys, are programmable *except*:

CLP (clear program)	***SIZE*** (number of storage registers)
[←] (correction)	[PRGM] (Program mode key)
[BST] (back step)	[GTO] [.] (go to line number)
[SST] (single step)	[CATALOG] (catalog list)
DEL (delete program lines)	***ON*** (continuous power)
[ASN] (assign)	[ON] (power on key)
[USER] (User mode key)	***COPY*** (copy program)
PACK (pack program memory)	[GTO] [.] [.] (go to end of program memory)

Pressing [■] [GTO] [.] [.] results in the following:

- If the last instruction in program memory is not an ***END*** instruction, the calculator inserts an ***END*** instruction at the end of the last program. This separates the last program in memory from the program to be added afterward.
- The calculator is positioned at the end of program memory (i.e., after the last program in memory, at the beginning of unused memory).

- Program memory is packed. This means that program instructions are moved up into unused bytes, interspersed within program memory, that may have resulted from inserting or deleting instructions.
- If the calculator is in Program mode, it displays ***00 REG*** followed by two (or three) digits indicating the number of unused registers in memory.

After adding an instruction at the end of program memory, you can determine how many memory registers remain unused by pressing [SST]. The calculator will then display ***.END. REG*** followed by the number of unused registers. Pressing [SST] again will set the calculator to line 01 of the current program, enabling you to review the entire program using [SST]; pressing [■] [BST] will set the calculator back to the last line of the current program, enabling you to continue adding instructions.

The number of unused registers available for program instructions may be increased in several ways:

- Execute ***PACK*** to pack program memory. This may make new registers available if instructions have been inserted or deleted, or if key assignments (of functions or programs listed in catalog 2 or 3) have been canceled,* since the last packing.
- Execute ***SIZE*** and specify a smaller number of data storage registers than are currently allocated.
- Delete complete programs (by executing ***CLP***).
- Delete single or multiple program lines, then execute ***PACK***.
- Cancel key assignments of functions or programs listed in catalog 2 or 3,* then execute ***PACK***.
- Add a memory module, then execute ***SIZE*** and specify the appropriate number of data storage registers. (Refer to Expanding Main Memory, page 20.)

If there are not enough bytes remaining in program memory to store an instruction being added, the calculator will pack program memory and display ***TRY AGAIN***. If the calculator displays ***TRY AGAIN*** after the instruction is keyed in again, additional instructions cannot be entered until registers are made available as described above.

A program loaded at the end of program memory (i.e., a program keyed in after [■] [GTO] [•] [•] is pressed) need not be terminated with an ***END*** or ***RTN*** instruction. A special ***END*** instruction is always maintained at the end of program memory, occupying the last three bytes of the last register before the unused registers.** It cannot be deleted, and instructions cannot be inserted after it. This ***END*** instruction appears as ***.END.*** when displayed.

---

*Canceling a key assignment will make an additional register available only if there had been an odd number of key assignments in effect. (Page 33 describes how key assignments consume memory registers.)

Because of this, there may be up to six unused bytes available for instructions between the last instruction in program memory and the permanent ***END* instruction. These bytes are in addition to those in the unused registers (seven bytes per register), the number of which appears in the calculator display following ***.END. REG*** or ***00 REG***.

Running a Program

To run a program, the HP-41C is first switched out of Program mode, then the calculator is initialized (i.e., data are placed in the X-register, storage registers, etc.). A program may then be run in a number of ways:

- By executing the program using [XEQ] followed by the label name (the Alpha mnemonic) of the program.
- By using [ASN] to assign the program to a key, then pressing that key in User mode.
- By positioning the calculator to the beginning of the program and pressing [R/S] or [SST] . [R/S] begins automatic execution with the current line of program memory. [SST] executes the instruction in the current line of memory and advances the calculator to the next program line.

When a program is run, the instructions in program memory are executed until an ***END*** (or ***RTN***) instruction is executed or until the program is halted. During execution, the **PRGM** annunciator and the label execution indicator (}-) appear in the display. Each time the program executes a program label, the }- moves across the display one position to the right. After reaching the last position on the right, }- resets to the left side of the display and continues. The }- will disappear from the display after a program executes a ***VIEW*** or ***AVIEW*** instruction. The contents of the register being viewed will remain in the display until the next ***CLD*** *(clear display)*, ***VIEW***, or ***AVIEW*** instruction is executed. ***CLD*** will return the }- to the display.

Program Components

Program Lines

In Program mode, the HP-41C display is set to one line of program memory at a time. Each line contains a complete instruction consisting of: 1) a function, 2) an Alpha string of up to 15 characters, or 3) a complete number (up to 10 digits, or up to 10 digits plus a two-digit exponent of 10).*

Lines are created automatically as instructions are loaded in Program mode. Each line is assigned a number to indicate its position within the program. Each separate program in the HP-41C has its own set of line numbers.

```
01 LBL"AREA
02 X↑2
03 PI
04 *
05 END
01 LBL"ZOT
02 "ANS =
03 ARCL 00
04 END
```

If a function is to be entered into a program line using its display-execution name (rather than by pressing the corresponding function key), [XEQ] must be pressed before the function name is keyed in, just as in Normal mode. If [XEQ] is not pressed first, the Alpha characters will not be recognized as a function name; instead, they will be treated as Alpha data and will be entered into the ALPHA register when the program line is executed.

*As in Normal mode, if a number consisting of 9 or 10 digits plus an exponent of 10 is to be entered into a program line, a decimal point must be keyed in before the ninth digit.

In the Program mode display of a program line, the symbol ᵀ indicates that the characters following comprise an Alpha string or (if preceded by ***XEQ***) a function name or (if preceded by ***LBL***) a global Alpha label.

Labels

Labels are placed in program memory by pressing ■ [LBL] in Program mode, followed by the desired label designator. Labels are used to name complete programs and to delimit routines and subroutines within a program. HP-41C labels can consist either of digits (*numeric* labels) or Alpha characters (*Alpha* labels). Furthermore, they are considered to be either *local* labels or *global* labels, in accordance with the different ways these labels are searched for (which are described under Label Searches, page 46).

Numeric Labels. Numeric labels are local labels consisting of any two digits, and are of two kinds:

- Labels 00 through 14 are short form numeric labels. They use only a single byte of program memory.
- Labels 15 through 99 require two bytes of program memory.

Local Alpha Labels. Used as program labels, the single Alpha characters A through J and a through e are local labels. These single letters should not be used to name complete programs—they are more useful when used inside programs. Local Alpha labels require two bytes of program memory.

Global Alpha Labels. Global Alpha labels may consist of any combination of up to seven Alpha characters (including digits) except the comma (,), period (.), and colon (:). Single-letter Alpha labels A through J and a through e are considered to be local labels, not global labels. In the Program mode display of a global Alpha Label, the label is preceded by ᵀ. Global Alpha labels require four bytes of program memory for the [LBL] instruction plus one additional byte for each character in the label itself.

The calculator treats global labels differently than local labels in several ways:

- A global Alpha label can be easily accessed from any point in program memory.
- Global Alpha labels appear in the display of catalog 1.
- Only a program that begins with a global Alpha label can be assigned to a key for User-mode execution.

For these reasons, it is best to begin complete programs with a global Alpha label rather than with a local Alpha or numeric label.

Section 6

Program Editing

Positioning Within Program Memory

Positioning With [GTO] [•]

Whether or not the HP-41C is set to program mode:

- [■] [GTO] [•] followed by a global Alpha label positions the calculator to that label in program memory. The search for the label begins with the last global Alpha label and proceeds upward in memory, stopping at the first correct label encountered. If the specified label is not encountered, the display shows ***NONEXISTENT*** and the calculator is reset to its previous position within program memory.
- [■] [GTO] [•] followed by a three-digit number (**nnn**) positions the calculator to that line number in the program to which the calculator is currently set.

To position the calculator to line numbers of 1,000 or greater when memory has been expanded by the addition of memory modules, press [■] [GTO] [•] as for other line numbers, but then press [EEX] followed by the last three digits of the line number.

Specifying a line number for [■] [GTO] [•] that is larger than the highest line number within the current program simply sets the calculator to that highest line number.

Positioning With [RTN]

If the HP-41C is not in program mode, pressing [■] [RTN] *(return)* resets the calculator to the top (line 00) of the current program.

Positioning With Catalog 1

To position the calculator to a program line containing a global Alpha label or ***END*** instruction:

1. Press [■] [CATALOG] 1 to display the global labels and ***END*** instructions stored in program memory.
2. To slow the listing (if necessary), press any key other than [ON] or [R/S].
3. Press [R/S] to halt the listing at the desired global label or ***END*** instruction.
4. To display the next item or the previous item in the catalog listing (if necessary), press [SST] or [■] [BST].
5. Press [←].

Using the catalog 1 listing as described above is the only way the calculator can be positioned to a program written without any global Alpha labels. In the catalog listing, the ***END*** instructions for such programs are any that are not immediately preceded by a global Alpha label.

Single Step and Back Step

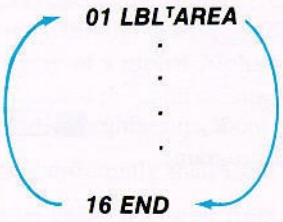
[SST] *(single step)* and [BST] *(back step)* permit the positioning of the calculator within a program (or catalog), and let the user view one line of a program at a time.

In Program mode:

- Each time [SST] is pressed, the next line of the current program is displayed.
- Each time ■ [BST] is pressed, the previous line of the program is displayed.

[SST] and [BST] operate only within the current program. Pressing [SST] when the calculator is set to the end of a program positions the calculator back at the beginning of that program. In a similar way, pressing ■ [BST] when the calculator is set to the top of a program positions the calculator around to the end of that program.

Pressing [SST] when the calculator is positioned here moves the calculator back to the beginning of this program.



Pressing ■ [BST] when the calculator is positioned at the top of the program moves the calculator to the end of this program.

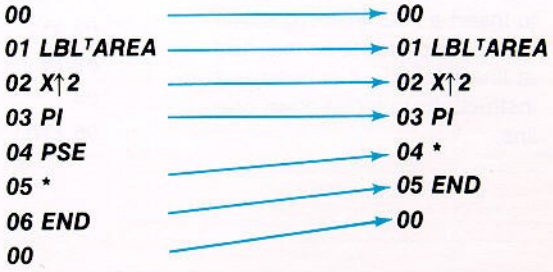
When [SST] is pressed, the next line of the current program is previewed; when [SST] is then released, that line is executed, permitting program execution one line at a time.

Deleting and Correcting Program Instructions

Deleting Instructions

Single Lines. To delete a single instruction, switch the calculator to Program mode, position the calculator to the program line containing the instruction to be deleted, then press ←. When a program line is deleted, the calculator moves to and displays the previous line in the current program, and the line numbers of all subsequent instructions in that program are reduced by one.

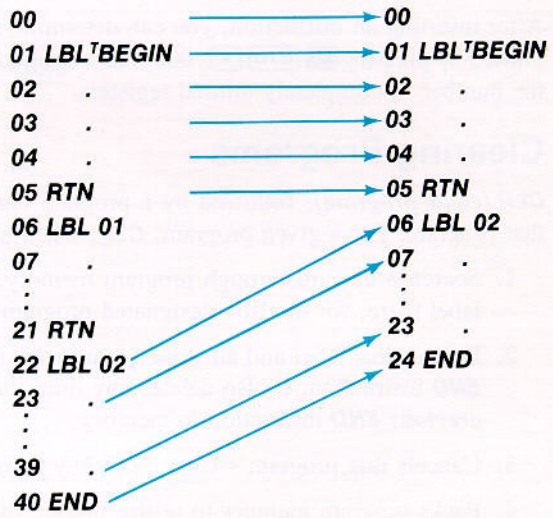
With the calculator set to line 04, a single press of ← deletes the instruction there, and subsequent instructions move up one line.



In general, when deleting a line at a time, begin with the last line to be deleted.

Multiple Lines. In Program mode, executing ***DEL*** *(delete lines)* followed by a three-digit line number **nnn** deletes the instruction in the current line of program memory plus those in all subsequent lines to a total of **nnn** or to, but not including, the next ***END*** instruction. The calculator then moves to and displays the previous line, and the line numbers of all subsequent instructions in the current program are reduced by **nnn**.

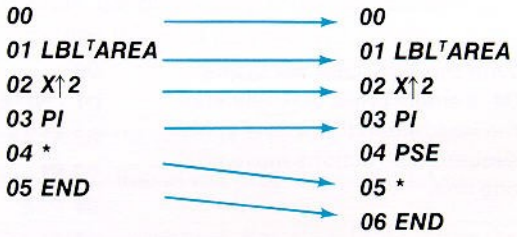
With the calculator set to line 06, executing ***DEL*** 016 deletes the instructions in 16 program lines, beginning with line 06, and subsequent instructions move up 16 lines.



Inserting Instructions

To insert an instruction in a program, set the calculator to the line number *preceding* that of the new instruction, then key the new instruction into memory. The calculator will display the new line, and the line numbers of all subsequent instructions in the current program will be increased by one.

With the calculator set initially to line 03, pressing the keys to insert a ***PSE*** instruction will cause the pause to be inserted at line 04 and all subsequent instructions to move down one line.



If there are not enough bytes remaining in program memory to store an instruction being inserted, the calculator will pack program memory and display ***PACKING*** followed by ***TRY AGAIN***. If the calculator displays ***TRY AGAIN*** after the instruction is keyed in again, additional instructions cannot be inserted until registers are made available (as described on page 36).

After inserting an instruction, you can determine how many registers remain completely unused by pressing ■ GTO • 000. The calculator will then display ***00 REG*** followed by the number of completely unused registers.

Clearing Programs

CLP *(clear program),* followed by a program name (i.e., a global Alpha label), clears that program. For a given program, ***CLP***, when executed:

1. Searches upward through program memory, beginning with the last global Alpha label there, for the first designated program label.
2. Deletes that label and all subsequent program lines down to and including the next ***END*** instruction. It also deletes any lines that may exist between the label and the *previous* ***END*** instruction in memory.
3. Cancels that program's User mode key assignment, if any.
4. Packs program memory to utilize unused bytes left by the cleared program.

For example, executing either ***CLP TEST 1*** or ***CLP TEST 2*** would change program memory

from this ...	to this ...
00	***00***
01 LBL^{T}ZOT	***01 LBL^{T}ZOT***
02 ^{T}ANS =	***02 ^{T}ANS=***
03 ARCL 00	***03 ARCL 00***
04 END	***04 END***
00	***00***
01 LBL^{T}TEST1	***01 LBL^{T}HEAT***
02 STO 01	***02 30***
03 RTN	
04 LBL^{T}TEST 2	
05 STO 02	
06 END	
00	
01 LBL^{T}HEAT	
02 30	

Executing ***CLP*** [ALPHA] [ALPHA] (i.e., without specifying a program name) clears the program to which the calculator is currently set.

To clear *all* programs in memory at once (as well as all other information maintained in the calculator's Continuous Memory):

1. Turn the calculator off.
2. Hold down the [←] key and press [ON] .
3. Release the [←] key.

Section 7

Program Interruptions

Using *STOP* and [R/S]

STOP

In Program mode, ***STOP*** can be loaded as an instruction by pressing the [R/S] (run/stop) key or by pressing [XEQ] [ALPHA] STOP [ALPHA] . When encountered later as an instruction by a running program, ***STOP*** is executed and the program halts, set to resume execution with the next line of program memory.

Keyboard Stops

If a program is running, pressing [R/S] stops the program. Although the ***STOP*** function can be assigned to other keys for execution in User mode, pressing those redefined keys (in Normal or in User mode) will not stop a running program. Only the [R/S] key itself (in addition to the [ON] key) can stop a running program. Furthermore, pressing [R/S] will stop a running program, in Normal or in User mode, even if another function has been assigned to the [R/S] key location. If a program is not running, pressing [R/S] starts the program running beginning with the current line in the program.

Using *PROMPT*

Executed in a running program, a ***PROMPT*** instruction causes the program to halt execution and display the contents of the ALPHA register.

Like ***AVIEW***, ***PROMPT*** only displays the contents of the ALPHA register; it does not switch the calculator to Alpha mode.

Using *PSE* (Pause)

When a running program executes a ***PSE*** *(pause)* instruction, execution is momentarily halted. During a pause—which lasts slightly less than 1 second—the calculator displays either the X-register or (if the calculator is in Alpha mode when ***PSE*** is executed) the ALPHA register. However, if—when ***PSE*** is executed—the display shows the contents of a register because a ***VIEW*** or ***AVIEW*** instruction has been executed, that display is maintained through the pause. Each time a ***PSE*** is executed, the **PRGM** annunciator blinks once.

During a pause (or string of pauses), the entire keyboard becomes active, and numeric or Alpha data can be entered into the calculator. Pressing any of the following keys during a pause causes the pause to be repeated: 0 through 9, [.] , [EEX] , [CHS] , [←] , [ALPHA] , [USER] , [■] , and all Alpha characters. If any other key is pressed during a pause, both the pause and program execution are terminated, and that key's function is executed.

Using *OFF*

When a program executes an ***OFF*** instruction, the calculator turns off.

Error Stops

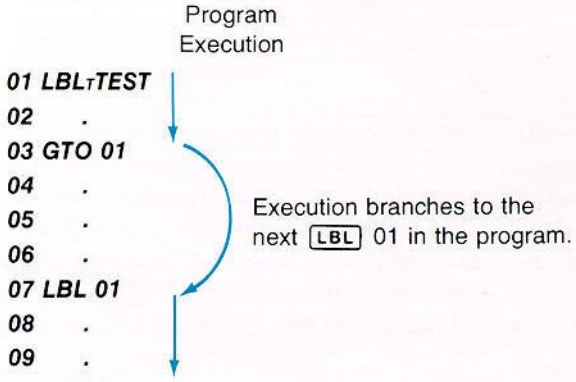
If the HP-41C attempts to execute an error-causing operation during a running program, execution halts and the HP-41C displays an error message. To see the line in the program containing the error-causing instruction, briefly set the calculator to Program mode. This clears the error message, as does pressing ⟵ . Flags 24 and 25 can be set to override error stops. Error messages and conditions are described in section 12 of this manual.

Section 8

Branching and Looping

Using [GTO] in a Program

Program execution can be made to branch to any local or global label by means of the [GTO] *(go to label)* instruction.



When the program encounters the ***GTO 01*** instruction, execution immediately halts and the calculator searches for ***LBL 01***, where execution resumes.

The label to which program execution branches (unless it is a local Alpha label) can be specified indirectly as well as directly. If the label specified in the indirect register is not found, or if the number or Alpha string in the indirect register is not a legal label (e.g., it is a number greater than 99 or a local Alpha label), program execution is halted at the line containing the [GTO] instruction, and the calculator displays ***NONEXISTENT***. Refer to Indirect Parameter Specification, page 8, for more information.

Label Searches

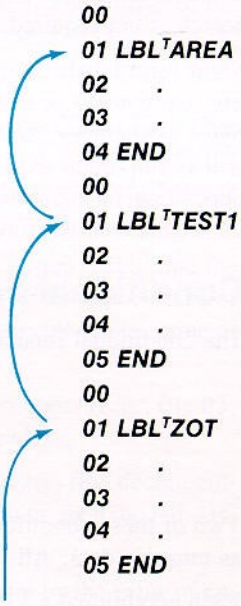
Global Label Searches

To find a global label, the calculator searches first through the global labels in program memory, then through the global labels in any application modules or peripheral devices that are connected to the calculator, and finally through the standard HP-41C functions. (The order of the search is indicated by the number of the corresponding catalog: catalog 1, catalog 2, then catalog 3.)

If a global Alpha label is comprised of the same characters as one of the standard HP-41C functions (e.g., ***ABS***, ***DEG***, ***SIN***), the calculator will execute the named program, rather than the corresponding function, when that label is specified after an [XEQ] or [GTO] instruction. However, if that label is not *already* contained in program memory or in a calculator extension (i.e., it is not listed in catalogs 1 or 2), the calculator will execute the standard HP-41C function.

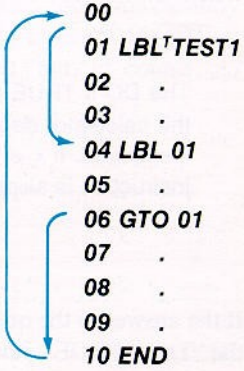
Within program memory, the search begins with the last global label and proceeds upward through memory, skipping all instructions except the global labels, until the specified label is found or until top of program memory is reached. If the label is found, execution begins there and proceeds downward in the usual manner.

If the specified label is not found (i.e., it is not listed in catalogs 1, 2, or 3), program execution is halted, the display shows ***NONEXISTENT***, and the calculator is reset to the same line in program memory as when the search began.



Local Label Searches

Searches for local labels occur only within the current program—that is, the program being executed or to which the HP-41C is set. To find a local label, the calculator first looks sequentially downward through the current program for the label. If the designated label is not found before reaching the end of the program (the ***END*** instruction), the calculator continues the search from the beginning of that same current program until it finds the first label with the specified name. If after a search the specified label has not been found, program execution is halted, the calculator is set to the same line of program memory as when the search began, and the display shows ***NONEXISTENT***.



Because of this label search process, local labels can be used any number of times, even in the same program.

A local label search can consume a significant amount of time, depending on the length of the current program. To minimize the search time, the calculator records the locations of most local labels during the first execution of a GTO ; so a local label search is not required for subsequent executions of that GTO . The exceptions are labels 00 through 14 ("short form" labels) when they are located more than 112 bytes from the GTO; for these labels, a label search is necessary during every execution of that GTO . If these labels are located 112 bytes or less before or after the GTO , however, a label search is not required.

Short form labels use only one byte of memory, and the corresponding GTO instructions use two bytes. Local Alpha and other numeric labels (15 through 99), however, use two bytes of memory, and the corresponding GTO instructions use three bytes. Therefore, if it is important to conserve program memory, short form labels should be used; but, depending on their location relative to the GTO instructions, use of these labels may increase execution time.

Conditional Functions

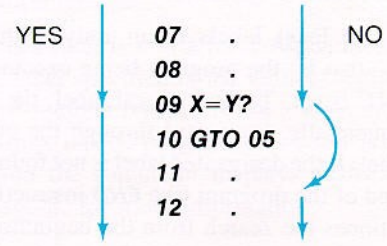
The conditional functions available for use on the HP-41C are:

x=y?	X≠Y?	x>y?	X<Y?	x≤y?
x=0?	X≠0?	X>0?	X<0?	X<=0?

Two of these conditionals, x=y? and X≠Y? can be used to compare Alpha data as well as numeric data. All the other conditionals compare only numeric data.

Each conditional asks a question; for example, the x=y? conditional asks if the data in the X-register of the stack is exactly equal to the data in the Y-register.

The DO if TRUE rule: If $x = y$, the calculator *does* the next instruction. If $x \neq y$, the next instruction is skipped.



If the answer to the question is YES, execution continues with the next line. This is called the "DO if TRUE" rule—if the conditional is TRUE, the calculator will DO the instruction in the next line of the program. If the answer to the question "Is the value in X equal to the value in Y?" is NO, execution skips one program line before resuming execution.

When any of these conditional functions is executed manually from the keyboard, the HP-41C displays the answer to the conditional question. If the condition is TRUE, the display shows **YES**; if the condition is false, the display shows **NO**.

Controlled Looping

The HP-41C has two functions for looping and the control of loops. These functions are [ISG] *(increment, skip if greater)* and ***DSE*** *(decrement, skip if equal)*. Both functions contain internal counters that allow loop control. These two functions interpret a control number in a special way. This control number may be placed into any data storage or stack register or the LAST X register. The register containing this control number should be specified in response to the prompt when [ISG] or ***DSE*** is executed. The format of the control number is:

iiiii.fffcc

iiiii is the current counter value. This value should consist of one to five digits. Each time [ISG] or ***DSE*** is executed, the current counter value **iiiii** is incremented or decremented by the value of **cc**. If [ISG] or ***DSE*** is used to implement a loop, the count of the number of passes through the loop is begun with the initial value of **iiiii**.

fff is the final counter value. If [ISG] or ***DSE*** is used to implement a loop, each time the instruction is executed the value of **iiiii** is compared to the value of **fff**. If the initial value of **iiiii** was zero, the value of **fff** is the number of times the loop will be executed. The value of **fff** must consist of three digits (e.g., 100, 009, etc.).

cc is the increment/decrement value. This value must consist of two digits (e.g., 01, 03, 25). If **cc** is 00, the calculator uses an increment value of 01 instead.

Each time ***DSE*** is executed, it examines the number in the control register, first decrementing **iiiii** by **cc**. It then tests to see if **iiiii** is equal to (or less than) **fff**. If it is, then the HP-41C skips the next line in program memory.

Each time [ISG] is executed, it first increments **iiiii** in the control register by the value of **cc**. If **iiiii** is then greater than **fff**, execution skips a line of the program and continues.

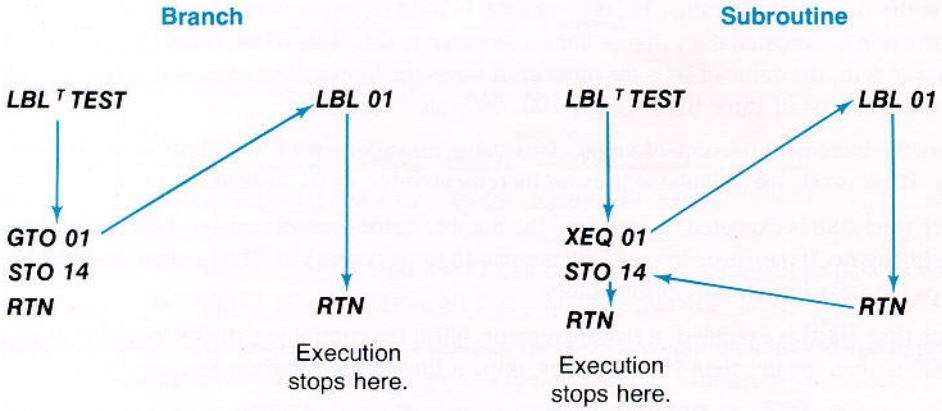
In a program, [ISG] or ***DSE*** instructions are typically placed within a loop immediately before a ***GTO*** instruction, to transfer execution out when **fff** is finally reached. If [ISG] or ***DSE*** is executed from the keyboard, the control register is altered, but, of course, no program memory lines are skipped.

Section 9

Subroutines

Subroutine Calls

A program instruction consisting of [XEQ] followed by a label is a subroutine call.* A subroutine call transfers execution to the desired label, just like a branch. But when a ***RTN*** or ***END*** instruction is then encountered, program execution is transferred back to the next line of program memory after the subroutine call.



Subroutine calls may be made to local Alpha or numeric labels within the current program, or they may be made to global Alpha labels anywhere in program memory.

The label to which program execution is transferred (unless it is a local Alpha label) can be specified indirectly as well as directly. If the label specified in the indirect register is not found, or if the number or Alpha string in the indirect register is not a legal label (e.g., it is a number greater than 99 or a local Alpha label), program execution is halted at the line containing the [XEQ] instruction, and the calculator displays ***NONEXISTENT***. Refer to Indirect Parameter Specification, page 8, for more information.

*The [XEQ] key executes subroutines just as the [GSB] key on other HP calculators with label addressing (such as the HP-67/97, the HP-19C/29C, and the HP-34C).

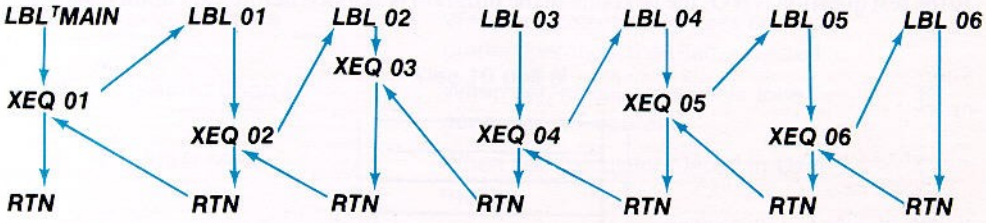
Subroutine calls perform label searches in the same way they are performed by a **GTO** instruction. If an **END** instruction is encountered before a specified local label is found, the search jumps to the beginning of the program and resumes. If a **RTN** instruction is encountered, however, the search proceeds beyond the **RTN**. For this reason, if there is more than one *locally*-labelled subroutine segment within a program, it is necessary to end each segment other than the last with a **RTN** instruction. Subroutines beginning with *global* labels can be ended with either an **END** or a **RTN** instruction. Subroutines located at the end of a program are terminated by the same **END** instruction that terminates the program; it is not necessary to terminate such a subroutine with its own **END** instruction.

```
00 LBL'DEMO
01   .
02   .
03 LBL 01
04   .
05   .
06   .
07 RTN
08 LBL 02
09   .
10 RTN
11 LBL 03
12   .
13   .
14 END
```

If a subroutine call specifies a local label that is not present within the current program, or a global Alpha label that is not present in program memory or in a calculator extension (i.e., it is not listed in catalogs 1 or 2), the calculator displays **NONEXISTENT**, the subroutine call is not performed, and execution is halted at the line containing the subroutine call.

Subroutine Limits

Subroutine branching is limited only by the number of **RTN**s (or **END**s) that can be held pending by the calculator. The HP-41C can hold up to six **RTN**s pending, so the calculator can return from subroutines up to six levels deep.



If any subroutine is executed manually from the keyboard, if [■] [RTN] is pressed, or if **SIZE** is executed, all pending **END** and **RTN** instructions are lost.

Section 10

Flags

A flag is a status indicator that can be in one of two states: "set" or "clear." Programs can use them to make decisions, just as it uses the conditional functions described in section 8.

The HP-41C contains a total of 56 flags. Of these, flags 00 through 29 are user flags; these can be set, cleared, and tested by the user. Flags 30 through 55 are system flags; these can be tested by the user, but not set nor cleared.

There are six flag functions. Each requires a two-digit flag number or indirect address for operation.

[SF] **nn** sets flag **nn**. (**nn** = 00 through 29.)

[CF] **nn** clears flag **nn**. (**nn** = 00 through 29.)

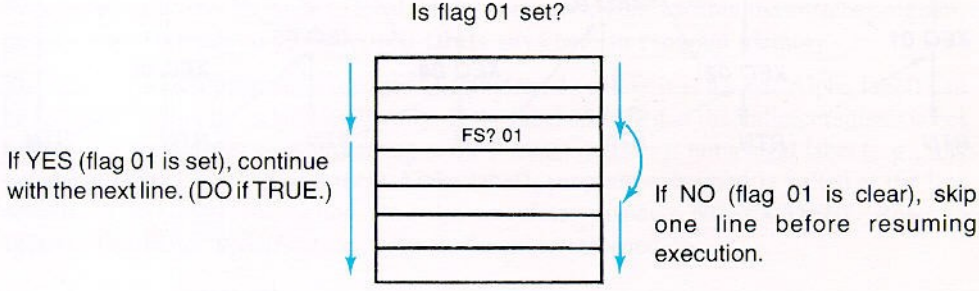
[FS?] **nn** asks "Is flag **nn** set?" (**nn** = 00 through 55.)

FC? **nn** asks "Is flag **nn** clear?" (**nn** = 00 through 55.)

FS?C **nn** asks "Is flag **nn** set?" Then flag **nn** is cleared by this instruction. (**nn** = 00 through 29.)

FC?C **nn** asks "Is flag **nn** clear?" Then flag **nn** is cleared by this instruction. (**nn** = 00 through 29.)

When a flag test function is executed in a program, if at that point in the program the answer to the test question is YES, the next line in the program is executed. If the answer to the test question is NO, the next line in the program is skipped before execution resumes.



Pressed from the keyboard, flag tests will display an answer of ***YES*** or ***NO*** to the test question.

Flag Number	Flag Name	Description	Status at Turn-On*
00-10	General Purpose	Status of flags 00 thru 04 appear in display.	M
11-20	Special Purpose	Status sometimes altered by calculator.	C
11	Automatic Execution	If set when calculator turned off, program execution begins when calculator turned on again.	C
12	Printer Double-Wide	If set, all characters are printed double-width.	C
13	Printer Lowercase	If set, all alphabetic characters are printed in lowercase.	C
14	Card Reader Overwrite	If set, data or program can be written on clipped (protected) card.	C
21	Printer Enable	If set, calculator assumes printer is present in system. Status matches flag 55 at turn-on.	†
22	Numeric Data Input	Set when numeric data entered from keyboard or (using HP 82153A Wand) bar code.	C
23	Alpha Data Input	Set when Alpha data entered from keyboard or bar code.	C
24	Range Error Ignore	When set, calculator ignores range error (i.e., a number generated that is greater than $9.999999999 \times 10^{99}$).	C
25	Error Ignore	When set, calculator ignores one improper operation, then flag is cleared.	C
26	Audio Enable	When set, calculator audible tone generator will operate.	S
27	User Mode	When set, calculator placed in User mode.	C
28	Decimal Point	When clear, numbers appear with comma as decimal point and dot as digit separator (for example, ***123.456.789,0***).	M
29	Digit Grouping	When clear, digit separators are omitted (e.g., ***123456789.0***).	M
30	Catalog	Used for operation of catalog. Always tests clear for user.	NA

Flag Number	Flag Name	Description	Status at Turn-On*
31-35	Peripheral	Used internally for operation of certain peripheral devices.	NA
36-39	Number of Digits	Used in combination to set number of displayed decimal digits in FIX, SCI, and ENG.	M
40-41	Display Format	Used in combination to select FIX, SCI or ENG display format.	M
42	Grads Mode	When set, calculator is in Grads mode.	M
43	Radians Mode	When set, calculator is in Radians mode.	M
44	Continuous On	When clear, calculator automatically turns off after 10 minutes of inactivity.	NA
45	System Data Entry	Used internally by HP-41C in data entry. Always tests clear for user.	NA
46	Partial Key Sequence	Used internally in function execution. Always tests clear for user.	NA
47	Shift Set	Used internally in shifted operations. Always tests clear for user.	NA
48	Alpha Mode	Set when HP-41C is in Alpha mode.	C
49	Low Battery	When set, battery power is low.	NA
50	Message	When set, display contains error/status message or ***VIEW/AVIEW*** display.	NA
51	SST	Used internally for single-step program execution. Always tests clear for user.	NA
52	Program Mode	Used to control Program mode. Always tests clear for user.	C
53	I/O	Used to determine if some peripheral device is ready for I/O (input or output). When set, device is ready.	NA
54	Pause	When set, ***PSE*** is in progress.	NA
55	Printer Existence	When set, a printer is attached to the HP-41C. Works in conjunction with flag 21.	†

*S = set; C = clear; M = maintained by Continuous Memory; NA = not applicable.

†Status of flags 21 and 55 match each time calculator is turned on. At turn-on, both flag 21 and 55 are set if a printer is in the system.

Section 11

Peripheral Devices

Names of programs and functions provided by application modules and other peripheral devices are displayed if ■ CATALOG 2 is pressed when the device is connected to the calculator. Names of *programs* are preceded in the catalog display by a ᵀ; programs can be copied into the calculator's memory and the copy then modified. Names of *functions* are not preceded in the catalog display by a ᵀ; functions cannot be copied into the calculator's memory and cannot be modified.

Both functions and programs can be executed, using the XEQ key, when the device is connected. If this XEQ instruction is contained in a program line, the device need not be connected to the calculator at the time the program is written. However, the form in which that instruction will subsequently be displayed depends on whether the device is connected at the time the program is written and at the time the program line is displayed:

- If the device is *not* connected when the program is written, instructions executing either a program or a function are always displayed as **XEQᵀ** followed by the function or program name.
- If the device *is* connected when the program is written:
  - Instructions executing a program are displayed—whenever the device is connected—as **XROMᵀ** followed by the program name.
  - Instructions executing a function are displayed—whenever the device is connected—as just the function name.
  - Instructions executing either a program or a function are displayed—when the device has been disconnected—as **XROM** followed by a two-digit device number and a two-digit program/function number.

For example, if the math application module and the peripheral printer are connected when a program is written:

Display of Instruction When Device Is Connected	Display of Instruction When Device Has Been Disconnected	
XROMᵀ TANH	**XROM 01,35**	Executes program **TANH**, which is number 35 in the catalog of device number 01, the math application module.

Display of Instruction When Device Is Connected	Display of Instruction When Device Has Been Disconnected	
PRX	***XROM 29,20***	Executes function ***PRX***, which is number 20 in the catalog of device number 29, the printer.

Instructions keyed into memory while a device *is* connected require two bytes of program memory. Instructions keyed into memory while a device is *not* connected require two bytes plus one additional byte for each character in the function or program name.

Programs provided by an application module or peripheral device can not only be executed just like programs stored in program memory; they can also be accessed (by pressing [■] [GTO] [•], then keying in the program name) and then reviewed (using [SST] and [BST]). However, these programs cannot be modified (i.e., program lines cannot be deleted and/or added) until the program has been copied from the module or device into the calculator's program memory. This copy can be done in either of two ways:

- Execute ***COPY***, then key in the name of the desired program.
- If the calculator is already positioned in the program (as a result of pressing [■] [GTO] [•] and keying in the program name), execute ***COPY***, then press [ALPHA] [ALPHA].

After this has been done:

1. The calculator searches for the specified name first in catalog 1, then in catalog 2. If the name is not found, or if a function rather than a program has been specified, the display will show ***NONEXISTENT***. If the program specified already exists in program memory, the display will show ***RAM***.
2. The calculator determines the length of the specified program and the amount of unused program memory.
3. If there are sufficient unused registers in program memory to accommodate the program, the program is copied into program memory. If not, the calculator will pack program memory and display ***PACKING*** followed by ***TRY AGAIN***. If the calculator displays ***TRY AGAIN*** after ***COPY*** is executed again on that program, the program cannot be copied into program memory until additional registers are made available (as described on page 36).

Attempting to modify a program from an application module or peripheral device before it has been copied into program memory will result in a ***ROM*** error message.

Section 12

Error and Status Messages

When an illegal operation is attempted on the HP-41C, the operation is not performed and an error/status message appears in the display. To clear the display, press [←]. If the error was caused during a running program, switching the calculator to Program mode will show the program line that attempted the meaningless operation.

Display	Meaning
ALPHA DATA	The HP-41C attempted to perform a numeric operation, such as addition or subtraction, on non-numeric data such as an ALPHA string.
DATA ERROR	The HP-41C attempted to perform an illegal operation. These errors are:
	[÷] where $x = 0$.
	[y^x] where $y = 0$ and $x \leq 0$, or where $y < 0$ and x is non-integer.
	[$\sqrt{x}$] where $x < 0$.
	[$1/x$] where $x = 0$.
	[LOG] where $x \leq 0$.
	[LN] where $x \leq 0$.
	LN1+X where $x \leq -1$.
	[$\cos^{-1}$] where $\lvert x \rvert > 1$.
	[$\sin^{-1}$] where $\lvert x \rvert > 1$.
	[STO] [÷] where $x = 0$.
	TONE where $\lvert x \rvert \geq 10$ or $x < 0$.
	MEAN where $n = 0$.
	OCT where $\lvert x \rvert >$ 1073741823 (decimal), or x is non-integer.
	DEC where x contains an Alpha string, 8 or 9, or x is non-integer.
	%CH where $y = 0$.
	[FIX], [SCI], [ENG] where absolute value of digits is ≥ 10 or is non-integer.
	FACT where $x < 0$ or x is non-integer.
MEMORY LOST	The Continuous Memory of the calculator has been cleared.
NONEXISTENT	The HP-41C has attempted to use a register that does not exist or is not currently allocated as a storage register.

An attempt was made to [ASN] or [XEQ] a function that does not exist.

An attempt was made to [ASN], [GTO], or [XEQ] an Alpha or numeric label that does not exist.

An attempt was made to [GTO] a line number that does not exist.

An attempt was made to execute a specific print function when the printer was not connected to the system.

NULL Keystroke was nullified by holding the key down for longer than about a half second.

PRIVATE Refer to the owner's handbook provided with the HP 82104A Card Reader.

An attempt was made to view a private program.

OUT OF RANGE A number has exceeded the computational or storage capability of the HP-41C.

Overflow = ± 9.999999999 99

SDEV where the standard deviation of x ($S_x = \sqrt{M/[n(n-1)]}$) or y ($S_y = \sqrt{N/[n(n-1)]}$) results in division by zero or the square root of a negative number. ($M = n\Sigma x^2 - (\Sigma x)^2$; $N = n\Sigma y^2 - (\Sigma y)^2$.)

FACT where $x > 69$.

PACKING Program memory is being packed.

TRY AGAIN As a result of a packing operation, the last keystroke sequence must be repeated. This could be an [XEQ], [ASN], [GTO] [•] [•], or when an attempt is made to insert an instruction into a program.

YES The answer to flag test when test is true. Also the answer to conditionals when relationship between x and 0 or y is true.

NO The answer to flag test when test is false. Also the answer to conditionals when relationship between x and 0 or y is false.

RAM An attempt was made to ***COPY*** into RAM (Random Access Memory: either internal memory or a memory module) a program already in RAM.

ROM An attempt was made to ***DEL***, ***CLP***, [←], or insert into a program that is currently in ROM (Read Only Memory: i.e., an application module).

Many peripheral devices that are plugged into the HP-41C contain their own unique error and status messages. These messages may also appear in the HP-41C calculator display. Refer to the literature accompanying the peripheral device for a description of its error/status messages.

Function Index

Function		Description	Prompts For	Stack Lift*	Indirect and LAST X†	Bytes‡	Flags Affected
Display Execution	Keyboard Execution						
	[■] **	Shift key.		N			47
+	[+]	Addition operator.		E	L	1	
−	[−]	Subtraction operator.		E	L	1	
*	[×]	Multiplication operator.		E	L	1	
/	[÷]	Division operator.		E	L	1	
1/X	[1/x]	Reciprocal.		E	L	1	
10↑X	[10^x]	Common antilogarithm.		E	L	1	
ABS		Absolute value.		E	L	1	
ACOS	[COS⁻¹]	Arc (inverse) cosine.		E	L	1	
ADV		Advance paper.§		E		1	21, 55
AOFF	[ALPHA] **	Alpha mode off.		E		1	23, 48
AON	[ALPHA] **	Alpha mode on.		N		1	
	[APPEND]	Append characters.	Character.	E		1	
ARCL	[ARCL]	Alpha recall. Recalls data into ALPHA register.	Register address.	E	I	2	
ASHF		Alpha shift left.		E		1	
ASIN	[SIN⁻¹]	Arc (inverse) sine.		E	L	1	

Function		Description	Prompts For	Stack Lift*	Indirect and LAST X†	Bytes‡	Flags Affected
Display Execution	Keyboard Execution						
ASN	[ASN]	Assign.	Function mnemonic; key location.	E		‡‡‡	
ASTO	[ASTO]	Store Alpha data in register.	Register address.	E	I	2	
ATAN	[TAN^{-1}]	Arc (inverse) tangent.		E	L	1	
AVIEW	[AVIEW]	Alpha view.		E		1	21, 50, 55
BEEP	[BEEP]	Beeper.		E		1	26
BST	[BST]	Back step.		E			
CAT	[CATALOG]	Catalog.	Catalog number (1, 2, or 3).	E	I		30
CF	[CF]	Clear flag.	**nn** flag number.	E	I	2	00-29
CHS	[CHS]	Change sign.		E		1	
CLA	[CLA]	Clear ALPHA register.		E		1	
CLD		Clear display.		E		1	
CLP		Clear program.	Program label.	E			
CLRG		Clear registers.		E		1	
CLΣ	[CLΣ]	Clear statistics registers.		E		1	
CLST		Clear stack registers.		E		1	
CLX	[CLX]	Clear X-register.		D		1	

Function		Description	Prompts For	Stack Lift*	Indirect and LAST X†	Bytes‡	Flags Affected
Display Execution	Keyboard Execution						
COPY		Copy program from application module or peripheral device into program memory.			E		
	←	Correction key.		N			
COS	COS	Cosine.		E	L	1	
D–R		Degrees to radians conversion.		E	L	1	
DEC		Octal to decimal conversion.		E	L	1	
DEG		Degrees mode.		E		1	42, 43
DEL		Delete program memory lines.	**nnn** lines to be deleted.	E			
DSE		Decrement, skip if equal.		E	I	2	
	EEX	Enter exponent.	**n**, exponent of 10.	E		1	
END		End program.		E		3	
ENG	ENG	Engineering notation.		E	I	2	36-41
ENTER↑	ENTER↑	Enter number in X-register into Y-register.		D		1	
E↑X	e^x	Natural antilogarithm.		E	L	1	
E↑X–1		Natural antilogarithm for arguments close to zero.		E	L	1	

Function		Description	Prompts For	Stack Lift*	Indirect and LAST X†	Bytes‡	Flags Affected
Display Execution	Keyboard Execution						
FACT		Factorial.		E	L	1	
FC?		"Flag clear" test.	**nn** flag number.	E	I	2	00-55
FC?C		"Flag clear" test and clear.	**nn** flag number.	E	I	2	00-29
FIX	[FIX]	Fixed point display.	**n**	E	I	2	36-41
FRC		Fractional portion of number.		E	L	1	
FS?	[FS?]	"Flag set" test.	**nn** flag number.	E	I	2	00-55
FS?C		"Flag set" test and clear.	**nn** flag number.	E	I	2	00-29
GRAD		Grads mode.		E		1	42
GTO	[GTO]	Go to label.	Label **nn** or Alpha label.	E	I	††	
	[GTO] [.]	Go to line number or Alpha label.	Line number **nnn** or Alpha label.	E	I		
	[GTO] [.] [.]	Go to end of program memory.		E			
HMS		Decimal hours to hours minutes, seconds conversion.		E	L	1	

Function		Description	Prompts For	Stack Lift*	Indirect and LAST X†	Bytes‡	Flags Affected
Display Execution	Keyboard Execution						
HMS+		Hours, minutes, seconds addition.		E	L	1	
HMS−		Hours, minutes, seconds subtraction.		E	L	1	
HR		Hours, minutes, seconds to decimal hours conversion.		E	L	1	
INT		Integer portion of number.		E	L	1	
ISG	ISG	Increment, skip if greater.	Register address **nn**.	E	I	2	
LASTX	LASTX	Recalls LAST X register contents to X-register.		E		1	
LBL	LBL	Program label.	Alpha label or **nn** label.	E		‡‡	
LN	LN	Natural logarithm.		E	L	1	
LN1+X		Natural logarithm for arguments close to one.		E	L	1	
LOG	LOG	Common logarithm.		E	L	1	
MEAN		Mean.		E	L	1	
MOD		Modulo (remainder).		E	L	1	
OCT		Decimal to octal conversion.		E	L	1	

Function		Description	Prompts For	Stack Lift*	Indirect and LAST X†	Bytes‡	Flags Affected
Display Execution	Keyboard Execution						
OFF		Power off.		E		1	
	[ON] **	Power on/off key.		N			11
ON		Power on (continuous).					44
P-R	[P-R]	Polar to rectangular conversion.		E	L	1	
PACK		Pack program memory.		E			
%	[%]	Percent.		E	L	1	
%CH		Percent of change.		E	L	1	
PI	[π]	Pi (3.141592654).		E		1	
	[PRGM] **	Switches into/out of Program mode.		N			52
PROMPT		Prompt.		E		1	21, 50, 55
PSE		Pause.		E		1	54
R↑		Roll up stack.		E		1	
R-D		Radians to degrees conversion.		E	L	1	
R-P	[R-P]	Rectangular to polar conversion.		E	L	1	
RAD		Radians mode.		E		1	43
RCL	[RCL]	Recall data from register into X-register.	Register address.	E	I	§§	

Function		Description	Prompts For	Stack Lift*	Indirect and LAST X†	Bytes‡	Flags Affected
Display Execution	Keyboard Execution						
RDN	R↓	Roll down stack.		E		1	
RND		Round.		E	L	1	
RTN	RTN	Return.		E		1	
SCI	SCI	Scientific notation.	**n**	E	I	2	
SDEV		Standard deviation.		E	L	1	
SF	SF	Set flag.	Flag number **nn**.	E	I	2	
Σ+	Σ+	Accumulations for statistics.		D	L	1	
Σ−	Σ−	Accumulation correction.		D	L	1	
ΣREG		Statistical register block specification.	**nn**	E	I	2	
SIGN		Sign of *x*.		E	L	1	
SIN	SIN	Sine.		E	L	1	
SIZE		Size of data storage register allocation.	**nnn**, number of data storage registers.	E			
SQRT	√x	Square root.		E	L	1	
SST	SST	Single step.		E			51
ST+	STO +	Storage register addition.	Register address.	E	I	2	

Function		Description	Prompts For	Stack Lift*	Indirect and LAST X†	Bytes†	Flags Affected
Display Execution	Keyboard Execution						
ST–	STO −	Storage register subtraction.	Register address.	E	I	2	
*ST**	STO ×	Storage register multiplication.	Register address.	E	I	2	
ST/	STO +	Storage register division.	Register address.	E	I	2	
STO	STO	Store numeric data in register.	Register address.	E	I	§§	
STOP	R/S	Stops program execution.		E		1	
TAN	TAN	Tangent.		E	L	1	
TONE		Tone.	**n**	E	I	2	26
	USER **	User mode key.		N			27
VIEW	VIEW	View register contents.	Register address.	E	I	2	21, 50, 55
X=0?	X=0?	"X=0?" conditional test.		E		1	
X≠0?		"X ≠ 0?" conditional test.		E		1	
X<0?		"X<0?" conditional test.		E		1	
X<=0?		"X≤0?" conditional test.		E		1	
X>0?		"X>0?" conditional test.		E		1	
X=Y?	X=Y?	"X=Y?" conditional test.†††		E		1	

Function		Description	Prompts For	Stack Lift*	Indirect and LAST X†	Bytes‡	Flags Affected
Display Execution	Keyboard Execution						
X≠Y?		"X≠Y?" conditional test.†††		E		1	
X<Y?		"X<Y?" conditional test.		E		1	
X<=Y?	[x≤y?]	"X≤Y?" conditional test.		E		1	
X>Y?	[x>y?]	"X>Y?" conditional test.		E		1	
X<>		Exchange X- and any register.	Register address.	E	I	2	
X<>Y	[x⇄y]	Exchange X- and Y-registers.		E		1	
XEQ	[XEQ]	Execute.	Numeric or Alpha label, or function name.	E	I	***	
X↑2	[x²]	Square.		E	L	1	
Y↑X	[yˣ]	Exponential.		E	L	1	

* E = enabling; D = disabling; N = neutral.

† I = parameter required can be specified indirectly; L = saves *x* in LAST X register.

‡ Number of program memory bytes occupied by function. No entry indicates that the function is not programmable. Numeric data occupies one byte for each digit in the number, plus another byte for each [•], [CHS], and [EEX] keyed in with the data. Alpha data occupies one byte for each character in the Alpha string, plus one additional byte for the entire string.

§ Valid only if printer is connected.

** Functions and programs cannot be assigned to this key location.

†† If **nn** is 00 thru 14, uses 2 bytes. If **nn** is 15 thru 99, uses 3 bytes. If local Alpha label is specified, uses 3 bytes. If global Alpha label is specified, uses 2 bytes plus 1 additional byte for each character in label. If parameter is specified indirectly, uses 2 bytes.

‡‡ If **nn** is 00 thru 14, uses 1 byte. If **nn** is 15 thru 99, uses 2 bytes. If local Alpha label is specified, uses 2 bytes. If global Alpha label is specified, uses 4 bytes plus 1 additional byte for each character in label.

§§ If **nn** is 00 thru 15, uses 1 byte. If **nn** is 16 thru 99, uses 2 bytes. If register is specified indirectly, or if X, Y, Z, T, or LAST X register is specified, uses 2 bytes.

*** If numeric or local Alpha label is specified, uses 3 bytes. If global Alpha label or function name is specified, uses 2 bytes plus 1 additional byte for each character in label or function name. If label is specified indirectly, uses 2 bytes.

††† Can compare Alpha data as well as numeric data.

‡‡‡ Assignments of functions and programs listed in catalogs 2 and 3 to key locations consume one register (seven bytes) for each odd-numbered assignment made. For example, the first assignment consumes one register, the second assignment no additional space, the third assignment another full register, etc. Assignments of programs that are stored in program memory (i.e., those listed in catalog 1) do not require additional space; the assignment is stored with that program's label.

Index

Page numbers in bold type indicate primary references; page numbers in regular type indicate secondary references.

A

Address, register, **8**, 9
Allocation, main memory, **18**
Alpha data, 24, **29**, 30, 37, 48
ALPHA DATA message, 29, **57**
ALPHA display annunciator, **6**
Alpha entry, 15, **27**, 28
Alpha execution, **13-15**, 37
[ALPHA], **6**, 8, 13, 27, 31, 32, 44
Alpha keyboard, 13, **26**, **Inside Back Cover**
Alpha mode, 6, 18, **26-30**, 31
ALPHA register, 15, 18, **26-30**, 44
ALPHA register, clearing, **28**
ALPHA register, displaying, **27**
ALPHA register, recalling data into, **29**
Alpha string, 18, 24, **26**, 28, 29, 37, 38
Annunciators, status, **6-7**
Appending characters, **28**, 30
AON, 27, 28
[APPEND], **28**
[ARCL], **29-30**
ASHF, **29**
[ASN], **31**, 35, 37
ASN _, 26, **31**
Assignments of functions and programs, **31-34**, 42
Assignments of local Alpha labels, **33**
Assignments, key, 18, **31-34**, 36, 42
[ASTO], 27, **29**
AVIEW, 15, **27-28**, 37, 44, 54

B

BAT display annunciator, **7**
Branching, **46-49**
[BST], **13**, 27, 35, **36, 39, 40**, 56
Bytes, program memory, **24**, 56, 59-68
Bytes, unused, **25**, 36

C

[CATALOG], **13**, 18, 35, **39**, 55
Catalog 1, **12**, 13, 18, 33, 38, **39-40**, 46, 47
Catalog 2, **12**, 33, 36, 46, 47, 55
Catalog 3, **12**, 33, 36, 46, 47
Catalogs, **12-13**, 32
Catalogs, displaying, **13**
Characters, appending, **28, 30**
[CF], **52**

C

[CHS], 24, 44
[CLA], 15, 27, **28**
CLD, **15, 16, 28,** 37
Clearing the ALPHA register, **28**
Clearing Continuous Memory, 16, **23**, 43
Clearing data storage registers, **16**
Clearing error messages, **15**
Clearing programs, **42-43**
Clearing the display, **15**
CLP, 25, 26, 35, 36, **42-43**
CLRG, **16**
[CLΣ], **15**
Conditional functions, 29, **48**
Continuous Memory, 16, **17**, 23, 33, 43
Control number, [ISG] and ***DSE***, **49**
Controlled looping, **49**
COPY, 35, **56**
Correction [←] key, **13, 15**, 16, **23, 28**, 35, **39, 41**, 44, 57

D

Data, Alpha, **24, 29, 30, 37**, 48
DATA ERROR message, **57**
Data, numeric, **24**, 29, 30, 48
Data, recalling, **15-16, 29-30**
Data storage registers, **15-16, 18-20, 23**
Data storage allocation, **18-23**
Data, storing, **15**
Data, storing in program memory, **37**
Decimal point, **8, 15**, 24, **37**, 44
DEL, 35, **41**
Deleting program lines, 36, **41**
Deleting programs, 36, **42-43**
Digit entry, **15**
Display annunciators, **6-7**
Display editing and clearing, **15**
Display execution, 5, **13-15**, 37
Display, label execution indicator in, 16, 28, **37**
Displaying ALPHA register, **27**
DSE, **49**

E

Editing Alpha strings, **28**
Editing a program, **39-43**
Editing the display, **15**
[EEX], 24, **39**, 44

E

END, 12, 18, 20, **35, 36, 37**, 39, 40, **41, 42, 47, 50, 51**
.END., **24, 36**
Error messages, 15, **57-58**
Error stops, **45**
Expanding main memory, **20-23**
Extended registers, **8**

F

FC?, **52**
FC?C, **52**
Flag display annunciators, **7**
Flags, **52-54**, 59-68
FS?, **52**
FS?C, **52**
Function names, 13, 38, 55
Function null, **7**, 33
Function preview, **7**, 34
Functions, assigning to keys, **31-34**
Functions, conditional, 29, **48**

G

Global Alpha label, 10, 12, 13, 31, **38**, 39, **46-47**
GRAD display annunciator, **7**
GTO, 26, 49, 50
GTO, **46, 48**
GTO ·, 35, **39**, 56
GTO · ·, 18, 20, 25, **35-36**, 42

I

IND __, **8**
Indirect parameter specification, **8**, 59-68
Indirect register, **8**, 50
Instructions, deleting, 36, **41**
Instructions, inserting or adding, 36, **42**
Interruptions, program, 28, **44-45**
ISG, **49**

K

Key assignments, 18, **31-34**, 36, 42
Keyboard stops, **44**
Keyboard, Alpha mode, 13, **26**, 27
Keycodes, **32**

L

Label execution indicator, 16, 28, **37**
Label searches, 42, **46-48**, 56
Labels, 13, 31, 33, **38**, 46-48, 50
Labels, global Alpha, 10, 12, 13, 31, **38**, 39, **46-47**, 50
Labels, indirect specification of, **10, 50**
Labels, local Alpha, 31, 33, **38, 47-48**, 50
Labels, numeric, 10, 31, **38**, 50
Labels, short form, **38, 48**
LAST X register, 8, 15, 59-68
LBL, 26, **46**
Line numbers, program, 24, **37**
Lines, program, 24, 36, **37**
Loading a program, **35-36**
Local Alpha label, 31, 33, **38**, 50
Looping, **46-49**

M

Main memory, **18**
Main memory allocation, **18**
MEMORY LOST message, 23, **57**
Memory modules, 17, 18, **20-23**, 36
Memory, Continuous, 16, **17**, 23, 33, 43
Memory, main, **18**
Memory, positioning within program, 35, **39-40**
Memory, program, 18, 19, 23, **24-25**
Mode, Alpha, 6, 18, **26-30**, 31
Mode, Normal, 6, 15, 32
Mode, Program, 6, 26, 35, 36, 38, 41
Mode, User, 6, **31-34**, 37
Modes, calculator operating, **6**

N

NO message, 48, 52, **58**
NONEXISTENT message, 11, 15, 32, 47, 50, 51, 56, **57-58**
Nonprogrammable operations, **35**
Normal mode, 6, 15, 32
Null, **7**, 33-34
NULL message, 7, 31, 33, **58**
Numeric data, **24**, 29, 30, 48
Numeric label, 10, 31, **38**, 50

O

OFF, 27, **45**
ON, **35**
ON, **6**, 16, **23**, 27, 32, 35, 44
Operating modes, **6**
OUT OF RANGE message, **58**

P

PACK, **25**, 35, **36**
Packing, **25, 36**, 42, 56, **58**
PACKING message, 20, 42, **56**
Parameter specification, **7-12**
Parameter specification, indirect, **8-12**
Parameter specification, single-key, **8**
Partition, movable memory, 17, 19, 20, 21, 22, 23, 24
Pause, **44**
Peripheral devices, 12, **55-56**, 58
Positioning within program memory, 35, **39-40**
Preview, **7**, 33, 34
PRGM display annunciator, **6**, 37
PRGM, **6**, 32, 55
PRIVATE message, 58
Program interruptions, 28, 44-45
Program label, 13, 31, 33, **38**, 46-48, 50
Program lines, 24, 36, **37**
Program memory, 18, 19, 23, **24-25**
Program memory, positioning within, 35, **39-40**
Program mode, 6, 26, 35, 36, 38, 41
Program names, 33, 35, 55, 56
Program, loading a, **35-36**
Program, running a, **37**
Programs, assigning to keys, **31-34**, 42
Programs, clearing, 36, **42-43**
PROMPT, **44**

P

Prompt sign, **7, 15**, 27
Prompting, **7**
PSE, 28, **44**, 54

R

RAD display annunciator, **7**
RAM message, 56, **58**
RCL, **16**
Reassignments, key, 18, **31-34**, 36, 42
Recalling data, **15-16, 29-30**
Recalling data into the ALPHA register, **29-30**
REG message, **18**, 20, 36, 42
Register, ALPHA, 15, 18, **26-30**, 44
Register, indirect, **8**, 50
Register, LAST X, 8, 15, 59-68
Registers, clearing data storage, **16**
Registers, extended, **8**
Registers, data storage, **15-16, 18-20, 23**
Registers, unused, **18**, 24, **36**, 56
ROM message, 56, **58**
R/S, **13**, 27, **37, 39, 44**
RTN, 20, 36, **37, 50, 51**
RTN, **39**, 51
Running a program, **37**

S

Searches, global label, **46-47**, 56
Searches, label, 42, **46-48**
Storage registers, storing data in, **15-16**
Storage registers, viewing, **16**
Storing data, **15-16**
String, Alpha, 18, 24, **26**, 28, 29, 37, 38
Strings, Alpha, storing, **29**
Subroutine limits, **51**
Subroutine, executing ***SIZE*** while halted, **51**
Subroutines, **50-51**

T

Top row keys, **8, 33**
TRY AGAIN message, 20, **36**, 56, **58**

U

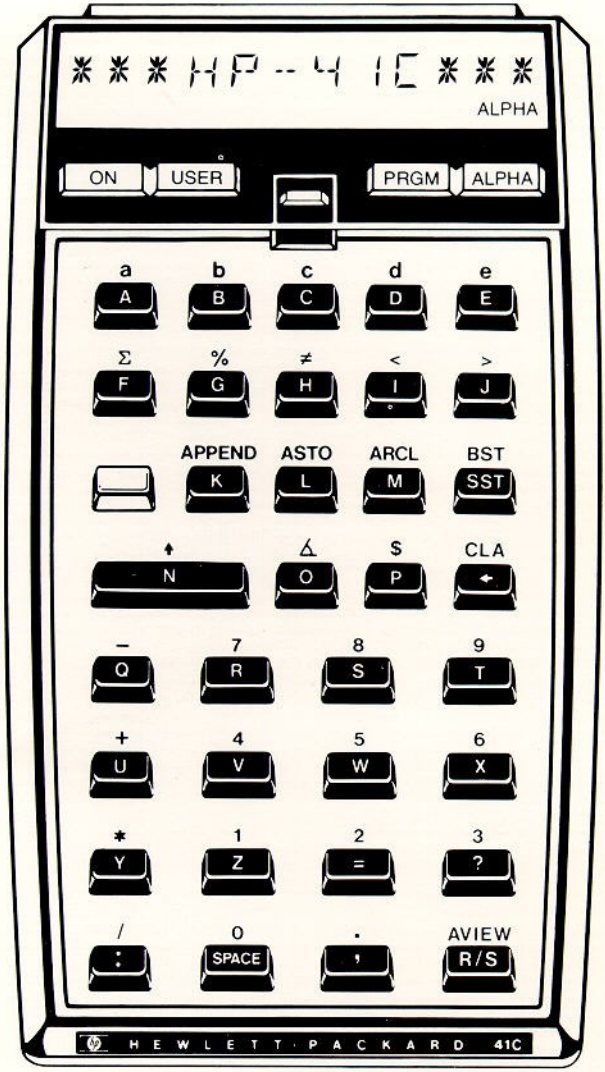
Unused bytes, **25**, 36
Unused registers, **18**, 24, **36**, 56
USER display annunciator, **6**
User mode, 6, **31-34**, 37
USER, **6**, 31, 32, 33, 35, 44

V

VIEW, 15, **16**, 28, 37, 44, 54
Viewing register contents, **16**

X

XEQ __, **13**, 26
XEQ, **13**, 33, **37, 50, 55**
XROM **55**



ALPHA Keyboard

1000 N.E. Circle Blvd., Corvallis, OR 97330

00041-90259

Printed in U.S.A.